



UNIVERSIDAD DE MÁLAGA



GRADO EN INGENIERÍA INFORMÁTICA

Sistema portátil para la monitorización de la calidad del agua

Portable water quality monitoring system

Realizado por
Christian Martos Reyes

Tutorizado por
Daniel Garrido Márquez
Cristian Martín Fernández

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO DE 2021



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA INFORMÁTICA

SISTEMA PORTÁTIL PARA LA MONITORIZACIÓN DE LA CALIDAD DEL AGUA

PORTABLE WATER QUALITY MONITORING SYSTEM

Realizado por
Christian Martos Reyes

Tutorizado por
Daniel Garrido Márquez
Cristian Martín Fernández

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2021

Fecha defensa: junio de 2021

Resumen

El Internet de las Cosas (IoT) es hoy en día un concepto que se aplica cada vez más a una mayor multitud de ámbitos. Ya en 2008 existían más “cosas u objetos” conectados a Internet que personas y cada vez se integran más sensores en objetos cotidianos facilitando la vida o solucionando necesidades de las personas.

El objetivo de este Trabajo de Fin de Grado ha sido proveer de un sistema de medición y representación de la calidad del agua para todas aquellas personas interesadas o preocupadas en conocer la calidad o salubridad del agua. Normalmente, las mediciones de la calidad del agua se tienen que realizar con multitud de aparatos de elevado coste y hay que anotar sus resultados de forma manual. Con este sistema se solventa este problema unificando todos estos sensores en un solo sistema integrado en Arduino, implementado con software libre y de bajo coste.

De forma resumida, el sistema está compuesto por un sistema Arduino con distintos módulos para la medición de los parámetros de turbidez, pH y temperatura del agua. Este utiliza una aplicación móvil Android a modo de *Gateway* para subir la información a la nube y mediante una aplicación web, los usuarios pueden visualizar todas las mediciones de forma comunitaria.

Palabras clave: IoT, Agua, Mediciones, Calidad

Abstract

The Internet of Things (IoT) is today a concept that is increasingly applied to a multitude of fields. Already in 2008 there were more “things or objects” connected to the Internet than people and more and more sensor are being integrate into everyday objects, making life easier or solving people’s needs.

The objective of this Final Degree Project has been to provide a system for measuring and representing water quality for all those interested or concerned in knowing the water situation. Usually, water quality measurements have to be carried out with a multitude of expensive devices and their results have to be recorded manually. With this system, this problem is solved by unifying all these sensors in a single system integrated in Arduino, implemented with free and low-cost software.

In summary, the system consists of an Arduino system with different modules for measuring turbidity, pH and water temperature parameters. This uses an Android mobile application as a Gateway to upload the information to the cloud and through a web application, users can view all the measurements in a community way.

Keywords: IoT, Water, Measurements, Quality

Índice

| | |
|---|-----------|
| Resumen | 1 |
| Abstract | 1 |
| Índice | 1 |
| Índice de ilustraciones | 1 |
| Índice de tablas | 1 |
| Introducción..... | 1 |
| 1.1 Motivación | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Estructura de la memoria | 3 |
| 1.4 Estudio previo de los parámetros del agua..... | 4 |
| 1.4.1 Parámetros físicos del agua..... | 4 |
| 1.4.2 Parámetros químicos..... | 8 |
| 1.4.3 Parámetros biológicos | 8 |
| 1.4.4 Parámetros elegidos..... | 8 |
| Tecnologías | 11 |
| 2.1 Hardware..... | 11 |
| 2.1.1 Arduino Mega | 11 |
| 2.1.2 Sensor ph DFROBOT SEN0161 | 12 |
| 2.1.3 Sensor Temperatura DFROBOT DFR0198..... | 13 |
| 2.1.4 Sensor de turbidez DFROBOT SEN0189..... | 14 |
| 2.1.5 Módulo Bluetooth DSD TECH HC-05 | 15 |
| 2.2 Software | 15 |
| 2.2.1 Android | 15 |
| 2.2.2 Android Studio..... | 17 |
| 2.2.3 Arduino | 18 |
| 2.2.4 Arduino IDE..... | 18 |
| 2.2.5 Firebase | 19 |
| 2.2.6 MagicDraw..... | 19 |
| 2.2.7 Angular | 20 |
| 2.2.8 Leaflet | 20 |
| Metodología..... | 21 |
| 3.1 Metodología de desarrollo | 21 |
| Análisis del sistema | 25 |
| 4.1 Descripción | 25 |
| 4.2 Requisitos funcionales..... | 25 |
| 4.3 Requisitos no funcionales..... | 27 |
| 4.4 Casos de uso | 27 |

| | |
|---|-----------|
| 4.4.1 Diagramas de casos de uso..... | 28 |
| 4.4.2 Especificación casos de uso | 29 |
| 4.4.3 Diagramas de secuencia | 33 |
| Diseño | 39 |
| 5.1 Diseño de la aplicación móvil | 39 |
| 5.1.1 Autenticación..... | 40 |
| 5.1.2 Registro..... | 41 |
| 5.1.3 Home..... | 41 |
| 5.1.4 Listado de dispositivos..... | 42 |
| 5.1.5 Recepción y subida de datos | 42 |
| 5.1.6 Visualización del mapa | 43 |
| 5.2 Diseño de la aplicación web | 43 |
| 5.3 Diseño de la base de datos | 44 |
| Implementación | 47 |
| 6.1 Implementación de la aplicación móvil..... | 47 |
| 6.1.1 Autenticación..... | 47 |
| 6.1.2 Registro..... | 49 |
| 6.1.3 Home | 50 |
| 6.1.4 Listado de dispositivos..... | 50 |
| 6.1.5 Recepción y subida de datos | 51 |
| 6.1.6 Visualización del mapa | 52 |
| 6.2 Implementación de la aplicación web..... | 52 |
| 6.3 Implementación de la base de datos | 53 |
| 6.4 Implementación de Arduino..... | 54 |
| 6.4.1 Conexiones hardware..... | 54 |
| 6.4.2 Implementación software | 56 |
| Pruebas | 59 |
| 7.1 Introducción a la fase de pruebas | 59 |
| 7.2 Pruebas realizadas | 59 |
| 7.3 Firebase Test Lab | 62 |
| 7.4 Firebase Crashlytics | 62 |
| Conclusiones y líneas futuras | 65 |
| 8.1 Conclusiones | 65 |
| 8.2 Líneas futuras | 66 |
| Referencias..... | 69 |
| Manual de Instalación..... | 71 |
| Requerimientos..... | 71 |
| Instalación de Arduino..... | 71 |
| Instalación de aplicación móvil | 72 |
| Instalación de aplicación web | 73 |
| Manuales de Usuario | 75 |
| Manual de Arduino | 75 |
| Manual de la Aplicación móvil | 75 |
| Manual de la Aplicación web | 81 |
| Coste del prototipo | 83 |

Índice de ilustraciones

| | |
|---|----|
| Ilustración 1 Diagrama esquemático de la estructura del sistema. | 2 |
| Ilustración 2 Placa Arduino Mega 2560 | 12 |
| Ilustración 3 Logo de DFROBOT | 12 |
| Ilustración 4 Sensor de pH | 13 |
| Ilustración 5 Sensor Temperatura..... | 14 |
| Ilustración 6 Sensor de Turbidez..... | 15 |
| Ilustración 7 Módulo Bluetooth HC-05..... | 15 |
| Ilustración 8 Sistemas Operativos en 2010 vs 2019 | 16 |
| Ilustración 9 Pila de Android | 16 |
| Ilustración 10 Ciclo de Vida de una actividad | 17 |
| Ilustración 11 Proceso de mejora en PSP | 22 |
| Ilustración 12 Ciclo de vida ágil | 23 |
| Ilustración 13 Casos de uso usuario móvil | 28 |
| Ilustración 14 Casos de uso usuario web | 28 |
| Ilustración 15 DS Autenticacion. EscHabitual | 33 |
| Ilustración 16 DS Autenticacion. EscAlternativo..... | 34 |
| Ilustración 17 DS Registro. EscHabitual | 34 |
| Ilustración 18 DS Registro. EscAlternativo | 35 |
| Ilustración 19 DS RecepcionDatos. EscHabitual..... | 35 |
| Ilustración 20 DS Recepción de Datos. EscAlternativo..... | 36 |
| Ilustración 21 DS Subida de Datos. EscHabitual | 36 |
| Ilustración 22 DS Subida de Datos. EscAlternativo | 37 |
| Ilustración 23 DS Visualizar medición | 37 |
| Ilustración 24 Diagrama de clases. Aplicación móvil..... | 40 |
| Ilustración 25 Diagrama de clases. Aplicación web | 43 |
| Ilustración 26 Modelo de datos NoSQL de Cloud Firestore | 45 |
| Ilustración 27 Servicio Authentication Firebase | 48 |
| Ilustración 28 Proveedores de acceso. Authentication..... | 48 |
| Ilustración 29 Gestión Usuarios. Authentication | 49 |
| Ilustración 30 Leaflet MakerCluster | 52 |
| Ilustración 31 Cloud Firestore | 53 |
| Ilustración 32 Diagrama de conexiones Arduino | 56 |

| | |
|---|----|
| Ilustración 33 Soluciones de calibración turbidez | 57 |
| Ilustración 34 Gráfica Voltios-NTU | 57 |
| Ilustración 35 Soluciones de calibración..... | 58 |
| Ilustración 36 Prueba Test Lab | 62 |
| Ilustración 37 Consola de Crashlytics | 64 |
| Ilustración 38 Instalación Arduino | 72 |
| Ilustración 39 Instalación Aplicaciones Desconocidas | 73 |
| Ilustración 40 Pantalla Inicio. Botón registro | 76 |
| Ilustración 41 Pantalla Registro..... | 76 |
| Ilustración 42 Pantalla de autenticación | 77 |
| Ilustración 43 Pantalla de Inicio..... | 78 |
| Ilustración 44 Pantalla selección dispositivo Arduino | 78 |
| Ilustración 45 Pantalla Recepción de datos | 79 |
| Ilustración 46 Pantalla envío de datos en progreso..... | 80 |
| Ilustración 47 Pantalla envío de datos satisfactorio..... | 80 |
| Ilustración 48 Mapa de mediciones. Aplicación móvil | 80 |
| Ilustración 49 Visualizar medición..... | 81 |
| Ilustración 50 Filtrar mediciones | 82 |

Índice de tablas

| | |
|---|----|
| Tabla 1 Dureza del agua medida por un sensor TDS, conductividad o dureza (Global Water a xylem brand, s.f.) | 6 |
| Tabla 2 CU Registro App..... | 29 |
| Tabla 3 CU Autenticación App..... | 30 |
| Tabla 4 CU Recepción de datos App | 30 |
| Tabla 5 CU Subida de datos App..... | 31 |
| Tabla 6 CU Visualizar mapa Web | 31 |
| Tabla 7 CU Filtrar por fecha Web | 31 |
| Tabla 8 CU Visualizar medición Web | 32 |
| Tabla 9 Caso de prueba 1 | 60 |
| Tabla 10 Caso de prueba 2..... | 60 |
| Tabla 11 Caso de prueba 3..... | 60 |
| Tabla 12 Caso de prueba 4..... | 61 |
| Tabla 13 Caso de prueba 5..... | 61 |
| Tabla 14 Caso de prueba 6..... | 61 |
| Tabla 15 Caso de prueba 7..... | 61 |
| Tabla 16 Caso de prueba 8..... | 61 |
| Tabla 17 Caso de prueba 9..... | 62 |
| Tabla 18 Coste del prototipo..... | 83 |

1

Introducción

1.1 Motivación

El agua es una de las sustancias más importantes. Sin ella no habría vida en la Tierra. Además de utilizarse para el consumo humano, el agua tiene otros muchos usos, estos son la limpieza, ocio, cultivos agrícolas y uso para ganadería, higiene personal, etc. Lo más importante es que el agua sea limpia y de calidad.

Es por ello por lo que, en 2010, la Asamblea General de las Naciones Unidas reconoció explícitamente “el derecho humano al agua y al saneamiento, donde todas las personas tienen derecho a agua suficiente, continua, segura, aceptable y físicamente accesible y asequible para uso personal y doméstico”. Un agua de baja calidad puede estar relacionada con la propagación de enfermedades y, además, suele desagradar a la población.

Debido a esta cuestión, muchas personas pueden considerar necesario disponer de la capacidad de controlar la calidad del agua presente en distintos entornos para poder compartir la información con otras personas también interesadas y crear una comunidad de voluntarios que tengan ganas de realizar esta labor.

La principal motivación es conseguir crear un sistema portátil y sobre todo de bajo coste con tecnologías del Internet de las Cosas (IoT) que sea capaz de

medir una serie de parámetros clave para determinar la calidad de un agua y almacenar esta información para su posterior consulta.

1.2 Objetivos

Este proyecto consistirá en la realización de un sistema basado en el Internet de las Cosas que realice distintas mediciones de la calidad del agua y que posteriormente se almacenen en el Cloud, la cuales pueden ser visualizadas en una aplicación web.

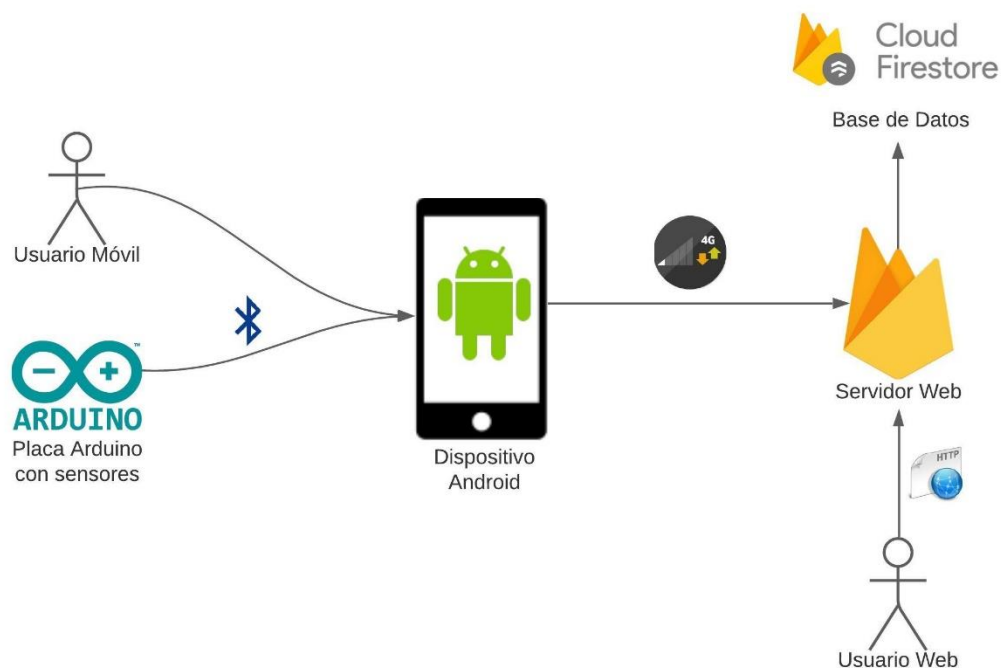


Ilustración 1 Diagrama esquemático de la estructura del sistema.

Como se puede observar en la ilustración 1, el sistema estará compuesto en primer lugar por un dispositivo empujado basado en Arduino con distintos sensores de medición de la calidad del agua.

En segundo lugar, este dispositivo se conectará mediante una tecnología inalámbrica a un teléfono móvil el cual usará una aplicación Android que será implementada y que se encargará de recibir los datos del dispositivo empujado y enviarlos al servidor Cloud.

El servidor Cloud llevará asociada una base de datos en la nube que se encargará de almacenar el histórico de los datos leídos.

Finalmente, se elaborará una aplicación web en la que se mostrará la información sobre las mediciones, lugar y otros aspectos que pudieran ser representados.

1.3 Estructura de la memoria

En el capítulo 1, se ha contextualizado el problema a resolver. Se ha descrito claramente de dónde surge la necesidad de este TFG y el dominio de aplicación. También se han especificado los objetivos generales y las funcionalidades del sistema a alto nivel. Finalmente, se ha realizado una investigación sobre los parámetros del agua.

En el capítulo 2, se enumeran y describen las tecnologías utilizadas (lenguajes de programación, *frameworks*, sistemas gestores de bases de datos, etc.) en el desarrollo del TFG.

En el capítulo 3, se hace una descripción de la metodología empleada en el desarrollo del TFG.

En el capítulo 4, se realiza un análisis técnico de los requisitos del sistema. Aquí se incluye la especificación de requisitos, casos de uso y diagramas de secuencia.

En el capítulo 5, se elabora un diseño del sistema, pasando por cada una de las partes estructurales del sistema.

En el capítulo 6, se lleva a cabo la implementación del sistema diseñado. Aquí el sistema es funcional.

En el capítulo 7, se ejecuta la elaboración de distintas pruebas para asegurarse que no existen errores funcionales y se han satisfecho los requisitos.

Por último, en el capítulo 8, se elaboran una serie de conclusiones y se proponen una serie de líneas futuras.

1.4 Estudio previo de los parámetros del agua

Además, se va a realizar un estudio para averiguar cuáles son los parámetros del agua adecuados para determinar su calidad. Estos serán de utilidad para la elección de los correspondientes sensores, es decir, el mayor grueso del hardware del sistema.

1.4.1 Parámetros físicos del agua

- **pH**

El pH o potencial de hidrógeno es una medida de acidez o alcalinidad. Es un valor numérico que cuantifica la concentración de iones de hidrógeno. El pH establece su propia escala, esta es logarítmica y va de 0 a 14, siendo 7 el pH neutro. Los pH inferiores a 7 se consideran ácidos y los pH superiores a 7 son alcalinos. (United States Geological Survey, s.f.)

El pH en el agua no debe ser excesivamente alto ni bajo, ya que un pH alto provoca sabor amargo, incrustaciones en las redes de distribución además de requerir mayores niveles de cloro adicional para la desinfección. Un pH bajo corroe o disuelve metales entre otras sustancias. Además, la cantidad de oxígeno en el agua aumenta a medida que aumenta el pH.

Es uno de los indicadores más importantes de la calidad del agua, ya que una alteración de los valores de pH indica una contaminación, ejemplos de esto serían las bacterias o la lluvia ácida, ambas ácidas o existencia de alcalinidad debido a contaminación industrial. El valor paramétrico correspondiente a un agua de buena calidad debe encontrarse entre 6,5 y 9,5 según el R.D. 140/2003, de 7 de febrero. (Boletín Oficial del Estado, 2003)

- **Conductividad**

La conductividad es una medida de la capacidad de pasar una corriente eléctrica a través del agua. La conductividad aumenta a medida que aumenta la salinidad, es decir, la cantidad de sales disueltas y otras sustancias químicas inorgánicas.

La conductividad se suele medir en $\mu S/cm$ (microsiemens por centímetro). (Fondriest Environmental, Inc., 2014) La escala de esta medida comienza en cero y un agua se “estima” que puede ser de mayor calidad cuanto más bajo sea el valor, sin embargo, muchas de las sustancias que pueden estar disueltas y que alteran el valor de la conductividad, no dicen nada sobre la calidad. Una alta conductividad debida a una elevada concentración de calcio es deseable en el agua, pero una pequeña concentración de arsénico, del orden de 20 μS , puede ser suficiente para producir un envenenamiento. (Omer, 2019)

Un concepto estrechamente relacionado con la conductividad es el de los sólidos disueltos totales (TDS). Representan la concentración total de sustancias, ya sean sales inorgánicas o materia orgánica, disueltas en el agua. Un alto nivel de TDS puede indicar que contaminantes pesados como el hierro, manganeso o arsénico, entre otros, pueden estar presentes en el agua. A nivel técnico no se puede certificar que estos niveles sean peligrosos para la salud, aunque pueden suscitar la realización un análisis químico del agua y comprobar si niveles de alguna de esas sustancias pudiese ser peligrosa. La unidad en la que se miden los sólidos disueltos totales es el miligramo por unidad de volumen en agua (mg/l), aunque también se utiliza las partes por millón (ppm).

A través de los sólidos disueltos totales podríamos establecer también una relación con la dureza del agua, que no es más que la concentración de compuestos minerales. La unidad de medida de la dureza son los grados hidrométricos franceses ($^{\circ}H F$ o simplemente $^{\circ}F$). (Facsa, 2017)

En general se puede medir la dureza del agua con un sensor de conductividad o con un sensor de TDS, ya que todas estas medidas están directamente relacionadas. La siguiente tabla muestra la dureza del agua medida por un sensor TDS, de conductividad o de dureza:

| TDS(ppm) | Conductividad ($\mu S/cm$) | Dureza ($^{\circ}F$) | Dureza |
|----------|------------------------------|------------------------|------------|
| 0-70 | 0-140 | 0-7 | Muy blanda |
| 70-150 | 140-300 | 7-15 | Blanda |

| | | | |
|---------|---------|-------|--------------------|
| 150-250 | 300-500 | 15-25 | Levemente dura |
| 250-320 | 500-640 | 25-32 | Moderadamente dura |
| 320-420 | 640-840 | 32-42 | Dura |
| >420 | >840 | >42 | Muy dura |

Tabla 1 Dureza del agua medida por un sensor TDS, conductividad o dureza (Global Water a xylem brand, s.f.)

- **Turbidez**

La turbidez es la medida de la claridad del agua. Una cantidad elevada de sedimentos en suspensión como arcillas u otras partículas en suspensión, pesticidas o metales pesados, provoca que la turbidez sea elevada y afecten a la calidad del agua, esto es debido a que las partículas suspendidas realizan una mayor absorción de la luz solar, llevando por consiguiente un aumento de la temperatura del agua y provocando una reducción de la concentración de oxígeno en el agua. Además, la turbidez en agua de calidad es estéticamente inaceptable, la cual hace que el agua tenga un aspecto no apetecible. Es por tanto que puede ser considerada uno de los parámetros más importantes junto con el pH. (Lenntech, s.f.)

La turbidez se mide en Unidades Nefelométricas de Turbidez (NTU) y según la OMS (Organización Mundial de la Salud), la turbidez del agua para consumo humano no debe ser superior en ningún caso a 5 NTU, recomendando que cuanto menor sea la turbidez mucho mejor, e idealmente estará por debajo de 1 NTU, concentración máxima permitida según el R.D. 140/2003 de 7 de febrero. (Boletín Oficial del Estado, 2003)

La turbidez se puede considerar como una aproximación a la concentración de sólidos totales en suspensión (TSS).

- **Temperatura**

La temperatura del agua es un parámetro que puede indicar el retardo o aceleración de la actividad biológica. Una temperatura elevada, por ejemplo, puede ser debida por un alto nivel de turbidez como antes se comentaba,

pudiendo provocar una disminución del oxígeno en el agua. La temperatura es un parámetro muy ligado a factores climatológicos del lugar donde se tome la muestra, hora del día, estación o incluso la profundidad del agua. Por ello se debe considerar una cronología y diferentes mediciones para establecer conclusiones. (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, s.f.)

- **Oxígeno disuelto**

El oxígeno disuelto, como su propio nombre indica, es una medida de la cantidad de oxígeno disuelto existente en el agua y se utiliza para dar una idea aproximada de la contaminación presente. A mayor concentración de oxígeno, mayor probabilidad de un entorno sano y estable. Existen otros factores como la salinidad, la temperatura o la altitud de los que depende la cantidad de oxígeno en un sistema acuático.

La unidad de medida de este parámetro es en partes por millón (ppm), equivalente a miligramos por litro (mg/L). Aunque este dato es muy variable, ya que depende de la especie, temperatura, etc., para una concentración de 5 a 6 ppm, hay oxígeno suficiente para la mayoría de las especies. Datos menos de 3 ppm indican que es dañino para el ecosistema y concentraciones menores a 2ppm es fatal para la mayor parte de las especies. (Universidad Complutense de Madrid, 2015)

- **Color, olor y sabor**

El color es un parámetro relacionado en cierta medida con la turbidez del agua. Los materiales descompuestos a partir de materia orgánica e inorgánica, como piedras, rocas, vegetación, etc. colorean el agua.

El color del agua se mide comparando una muestra de agua con soluciones de color estándar o con discos de vidrio coloreado. El color se clasifica en una escala de 0 (transparente) a 70 unidades de color. (Omer, 2019)

El olor y el sabor en el agua puede ser causado por materiales orgánicos o inorgánicos o gases disueltos. El valor numérico del olor o sabor se determina cuantitativamente midiendo un volumen de muestra A y diluyéndolo en un

volumen de muestra B de agua destilada sin olor de modo que el olor de la mezcla resultante sea simplemente detectable en un volumen total de mezcla de 200ml.

Como podemos observar, estos parámetros son análisis organolépticos realizados bajo interpretación humana, es decir, de determinación subjetiva, y por el momento no son fácilmente automatizables.

1.4.2 Parámetros químicos

Con respecto a los parámetros químicos del agua, principalmente se estudia la existencia de determinados elementos químicos como el aluminio, componente natural del agua, u otros elementos no tan deseables como el mercurio, metal pesado muy tóxico, pasando por elementos como el sodio, arsénico, plomo, etc. Estos parámetros se examinan mediante analíticas de laboratorio y por lo tanto quedan fuera de este estudio al no ser fácilmente detectables con sensores en tiempo real. (Analizar mi agua, 2019)

1.4.3 Parámetros biológicos

Por último, otras de las características que se miden son los parámetros biológicos del agua. Aquí se incluyen por ejemplo elementos biológicos como las algas u otros elementos orgánicos como los hongos, mohos o levaduras. Al igual que en el caso anterior, quedan fuera de nuestro estudio.

1.4.4 Parámetros elegidos

Tras analizar los distintos parámetros del agua, se han elegido los siguientes para ser medidos a través del sistema:

- pH. Este parámetro se ha seleccionado porque se trata de uno de los indicadores más importantes de la calidad del agua. Esto es debido a que una alteración de los valores de pH indica una contaminación en la mayoría de las ocasiones. Además como se muestra en el siguiente apartado, el sensor que realiza la medición de este parámetro tiene un coste aceptable en relación a la información que aporta.

- Turbidez. Es un parámetro cuya medición puede ser muy interesante, ya que aporta información relacionada con otros parámetros, como por ejemplo el oxígeno disuelto (cuyo sensor correspondiente es extremadamente caro). Además, es un valor muy a tener en cuenta, ya que un valor elevado de turbidez implica una alta cantidad de partículas en suspensión y, por consiguiente, un empobrecimiento de la calidad del agua.
- Temperatura. Este valor proporciona información que puede ser de utilidad para concluir que existe un problema en la calidad del agua. Esto es debido a que una elevada temperatura implica una reducción de la actividad biológica, como ocurre por ejemplo en los estancamientos de agua. El sensor que realiza su medición es bastante económico y muy preciso.

2

Tecnologías

Dentro de las tecnologías, podemos hacer una distinción entre las tecnologías que utilizaremos para el hardware y para el software

2.1 Hardware

2.1.1 Arduino Mega

Arduino Mega 2560 (Ilustración 2) es una placa basada en el microcontrolador ATmega2560. Tiene 54 pines de entrada/salida digital, 16 entradas analógicas, 4 puertos serie, un oscilador de cristal de 16MHz, una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio. Con respecto a la memoria, esta placa incluye una memoria flash de 256 KB de los cuales 8 KB son usados por el gestor de arranque. Incorpora una memoria SRAM de 8 KB y una EEPROM de 4 KB. Todo esto compone una de las placas más potentes y versátiles que podemos encontrar en Arduino a un precio moderado.



Ilustración 2 Placa Arduino Mega 2560

2.1.2 Sensor ph DFROBOT SEN0161

El sensor utilizado para la medición del pH del agua es el pH meter SEN0161 de DFROBOT (Ilustración 4). Este sensor tiene un precio que ronda los 25€, precio que comparado con otros sensores profesionales los cuales superan con creces los 100€, es una opción más que adecuada.

DFRobot se fundó a partir de una comunidad de fabricantes locales en 2008, siendo uno de los primeros en adoptar el hardware de código abierto, apuestan por crear productos de software y hardware innovadores y fáciles de usar, los cuales se convierten en los componentes estructurales de todo tipo de proyectos electrónicos enfocados a la comunidad de estudiantes.



Ilustración 3 Logo de DFROBOT

Este medidor incluye:

- Una sonda de pH, que realiza una medición analógica y se conecta mediante un conector BNC a un componente físico llamado módulo.
- Un módulo, también llamado sensor, interfaz o shield en inglés. En este caso, es un módulo de pH, que funciona como interfaz entre el electrodo de pH

y la placa Arduino Mega 2560. Este módulo se conecta con el cable suministrado a una entrada analógica de Arduino, de esta forma, la placa Arduino recibirá una tensión que estará directamente relacionada con el valor de pH.



Ilustración 4 Sensor de pH

El funcionamiento del sensor consiste en la medición de la diferencia de potencial que se crea entre dos electrodos situados en la sonda: un electrodo de referencia de plata/cloruro de plata y un electrodo de cristal PH, sensible a los iones de hidrógeno. La interfaz se encarga de modular la señal para que sea leída adecuadamente por el microcontrolador, en este caso la placa Arduino Mega.

2.1.3 Sensor Temperatura DFROBOT DFR0198

El sensor elegido para medir la temperatura es el DFR0198 (Ilustración 5) también de la marca DFROBOT. Se trata evidentemente de un medidor resistente al agua, cuyo rango de funcionamiento oscila entre los -55 °C y los 125 °C.

Este sensor está compuesto de:

- Una sonda DS18B20, cuyo encapsulado de fábrica es tipo TO-92, muy utilizado en la construcción de transistores. La presentación de esta sonda es dentro de un tubo de acero inoxidable resistente al agua.

- Adaptador de sensor de terminal DFR0055. Este adaptador se utiliza en sustitución de la resistencia de 4.7K Ohm entre el voltaje y el pin de señal que requiere la sonda para eliminar cualquier tipo de señal de ruido.

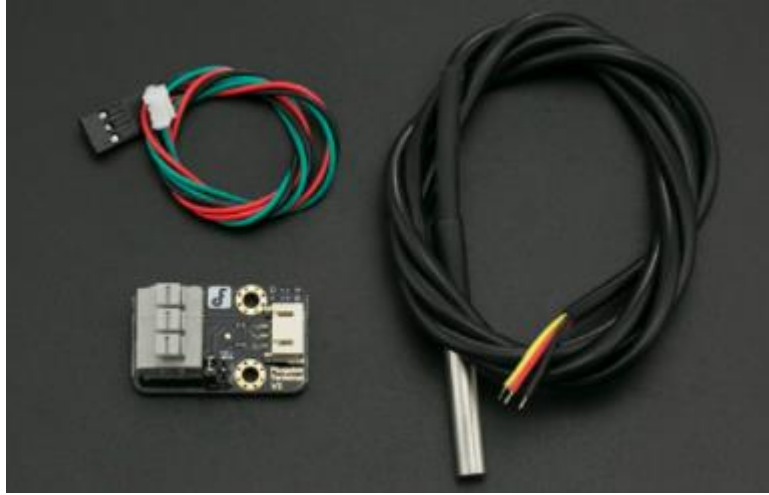


Ilustración 5 Sensor Temperatura

2.1.4 Sensor de turbidez DFROBOT SEN0189

El sensor elegido para detectar la calidad del agua midiendo los niveles de turbidez u opacidad es el SEN0189 (Ilustración 6) de DFROBOT. Este sensor utiliza luz para detectar partículas en suspensión midiendo la transmitancia de la luz y la tasa de dispersión, que cambia con la cantidad de sólidos suspendidos totales, es decir, a medida que aumentan los sólidos suspendidos totales, aumenta el nivel de turbidez del líquido.

Este sensor está compuesto de:

- Una sonda que realiza una medición analógica de la turbidez del líquido.
- Un módulo M019.00578 que permite alternar entre una señal analógica o una digital de salida, dispone de un potenciómetro y se encarga de trasladar la señal, sea del tipo que sea, a la placa Arduino eliminando en la medida de lo posible cualquier ruido.



Ilustración 6 Sensor de Turbidez

2.1.5 Módulo Bluetooth DSD TECH HC-05

El HC-05 (Ilustración 7) es un módulo Bluetooth 2.0 que permite añadir funciones de comunicación bluetooth a cualquier proyecto Arduino de una forma rápida. La configuración de fábrica de este módulo es en modo esclavo, es decir, solo recibe comunicaciones, pero se puede configurar en modo maestro para conectarse a otros dispositivos Bluetooth 2.0.

Utiliza el chip bluetooth convencional CSR BC417 y estándares del protocolo bluetooth V2.0 SPP (Serial Port Profile).

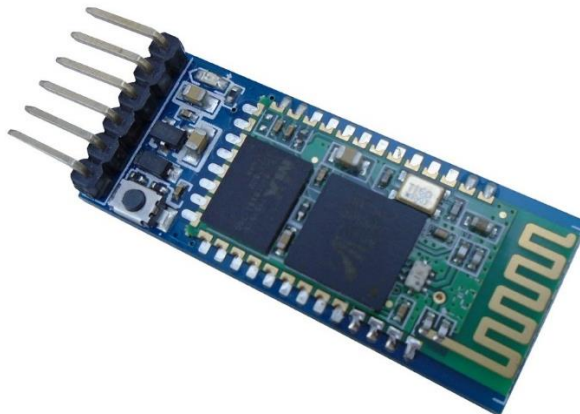


Ilustración 7 Módulo Bluetooth HC-05

2.2 Software

2.2.1 Android

Android es uno de los sistemas operativos móviles más populares en la actualidad. En la siguiente ilustración se puede observar la cuota de mercado de

smartphones por sistemas operativos, basado en unidades distribuidas, en 2010, y de nuevo estos datos en 2019. Se puede extraer la importancia que tiene Android en la actualidad del diagrama de la ilustración 8.

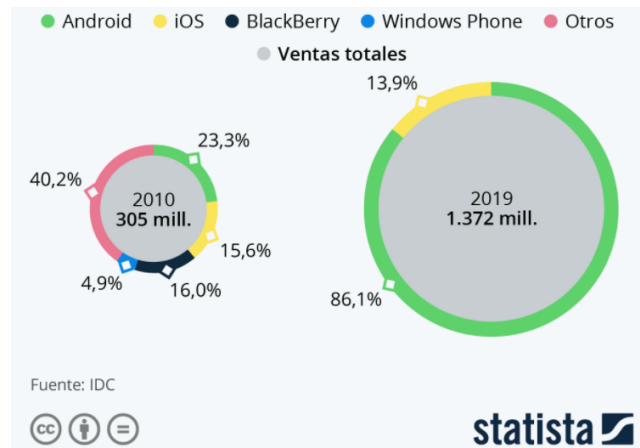


Ilustración 8 Sistemas Operativos en 2010 vs 2019

Android está basado en un núcleo Linux, en sus versiones 4.14, 1.19 o 5.4, y en otros softwares de código abierto. Fue presentado en noviembre de 2007, y el primer dispositivo Android comercial se lanzó en septiembre de 2008. Los paquetes de software en Android, aplicaciones o apps, usan el formato APK (Android application package) y usualmente se obtienen de tiendas de aplicaciones como Google Play Store.

La arquitectura de un sistema Android está basado en la colección de diferentes capas. Cada capa tiene un rol específico y un conjunto de funcionalidades, donde cada capa proporciona funcionalidad a la capa superior, estas capas vienen dadas por la pila de Android y son las mostradas en la ilustración 9.

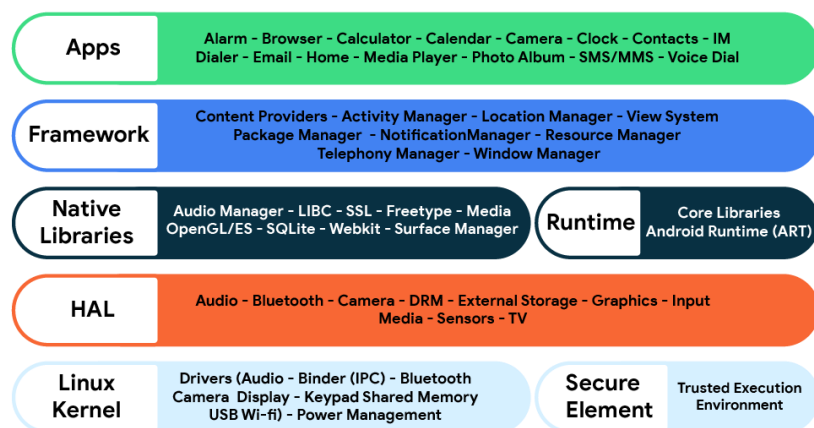


Ilustración 9 Pila de Android

Las aplicaciones, apps o paquetes de software en Android se desarrollan utilizando el kit de desarrollo de software de Android (SDK), como lenguaje de programación Kotlin o Java, acompañado a veces de lenguajes como C++. Estas aplicaciones tienen un componente clave, la clase Activity. A diferencia de otros paradigmas de programación donde el comienzo de la ejecución de código siempre es desde un `main()`, en el sistema Android esto se lleva a cabo desde una instancia de Activity donde se llevan a cabo invocaciones para ir navegando por las diferentes etapas del ciclo de vida de la actividad. En la ilustración 10 podemos observar una representación visual de este paradigma.

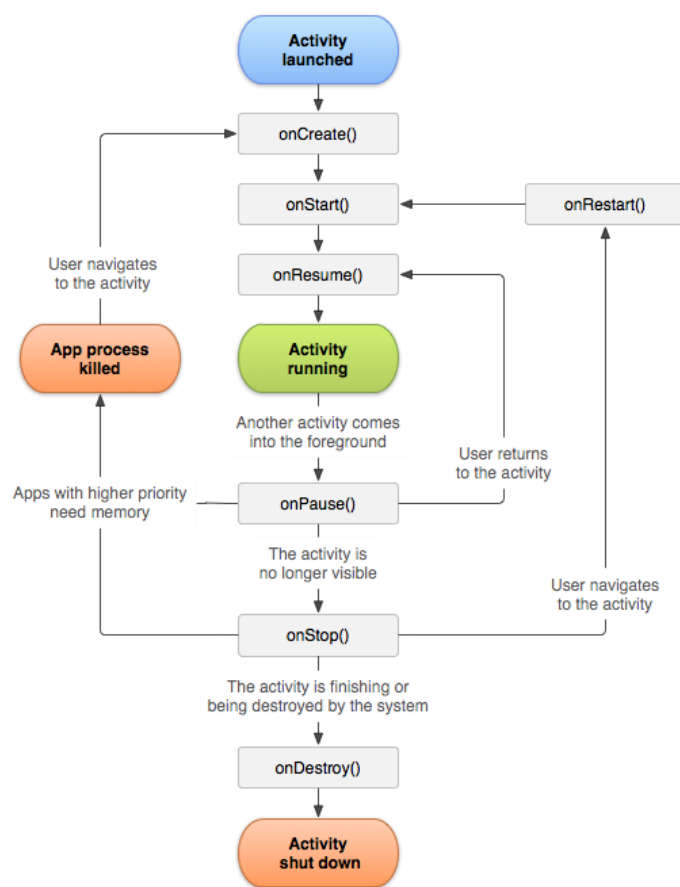


Ilustración 10 Ciclo de Vida de una actividad

2.2.2 Android Studio

Android Studio es el principal entorno de desarrollo integrado (IDE) para el desarrollo de aplicaciones Android. Basado en JetBrains IntelliJ IDEA, sustituyó en 2014 a IDE Eclipse. Gracias al SDK y a Android Studio, se incluyen un conjunto completo de herramientas de desarrollo, entre las que destacan un

depurador, bibliotecas de software, un emulador de teléfono basado en QEMU (Quick Emulator), documentaciones, código de muestra y tutoriales entre otros.

Una funcionalidad destacada de Android Studio es que ofrece un editor visual con la función de arrastrar y soltar cuando se trabaja con archivos XML, el cual renderiza el diseño o layouts en tiempo real.

2.2.3 Arduino

Arduino es una plataforma de electrónica de código abierto basada en la facilidad del uso del software y el hardware. Arduino tiene una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores...) cuyas aportaciones suman una gran cantidad de conocimiento accesible para ser de ayuda tanto a expertos como a principiantes. Algunas ventajas que aporta Arduino son:

- Económico: las placas Arduino son económicas en comparación con otros microcontroladores.
- Multiplataforma: el IDE de Arduino es compatible con Windows, Macintosh OS X y Linux.
- Entorno de programación simple y claro: el IDE de Arduino es accesible e intuitivo para principiantes y flexible para que los profesionales lo aprovechen.
- Software de código abierto y extensible: Arduino se publica en código abierto y se puede expandir a través de bibliotecas programadas en C++.
- Hardware de código abierto y extensible: los planos de todas las placas Arduino se difunden con licencia Creative Commons y cualquier diseñador de circuitos puede crear su propia versión de un módulo, ampliarlo o mejorarlo.

2.2.4 Arduino IDE

El Entorno de Desarrollo Integrado (IDE) de Arduino permite escribir y subir programas a las placas Arduino compatibles. Estos programas están escritos en C y C++ usando reglas especiales de estructuración de código, se les denomina bocetos o sketches y se guardan con la extensión de archivo .ino. Se trata de

una aplicación multiplataforma para Windows, macOS y Linux cuyo código fuente se publica bajo la Licencia General Pública GNU, versión 2.

2.2.5 Firebase

Firebase es una plataforma de Google ubicada en la nube para el desarrollo de aplicaciones web y móviles. Está integrada con Google Cloud Platform y ofrece sus servicios de forma gratuita hasta cierto nivel de demanda, momento en el cual se abonarán los servicios en función del uso. Firebase dispone de una gran cantidad de servicios, entre los que destacan:

- Google Analytics, solución de medición de análisis de aplicaciones que proporciona información sobre estadísticas relacionadas con el uso de la aplicación por parte de los usuarios.
- Firebase Authentication, servicio para autenticar usuarios mediante proveedores como Facebook, GitHub, Twitter y Google, o simplemente mediante el sistema de administración de usuarios que permite a los desarrolladores autenticar a los usuarios mediante correo electrónico y contraseña almacenadas en Firebase, siendo esta la opción elegida en este proyecto.
- Firebase Realtime Database, base de datos en tiempo real que proporciona a los desarrolladores una API para sincronizar los datos de la aplicación entre clientes y almacenarlos en la nube de Firebase.
- Cloud Firestore, sucesor de Firebase Realtime Database, el cual permite usar documentos y campos anidados en lugar de la vista de árbol proporcionada por el anterior. Se trata de una base de datos NoSQL flexible y escalable a la que pueden acceder las apps para iOS, Android y Web directamente desde los SDK nativos. En este proyecto se hará uso de esta base de datos.

2.2.6 MagicDraw

MagicDraw es una herramienta de modelado visual de sistemas, software, arquitectura y procesos de negocios compatible con el metamodelo UML 2

además de otros lenguajes o tecnologías de modelado como OCL (Object Constraint Language), SysML, UAF, UPDM, BPMN, ALF, BÚHO o TOGAF entre otros.

Diseñada para analistas de negocio o software, programadores, ingenieros de QA o encargados de documentar proyectos, esta herramienta de desarrollo dinámica y ligera facilita el análisis y diseño de sistemas de información o bases de datos orientadas a objetos.

2.2.7 Angular

Angular es un framework de desarrollo web modular y escalable basado en componentes web. Está basado en TypeScript, el cual es un lenguaje construido a partir de JavaScript, pero que está fuertemente tipado y además mejora la ayuda contextual. Angular es uno de los frameworks más utilizados y tiene la ventaja de ser soportado por Google, por lo tanto encaja perfectamente en el proyecto.

2.2.8 Leaflet

Leaflet es una de las bibliotecas JavaScript de código abierto para mapas interactivos más importante. Es la alternativa directa a Google Maps Platform cuyo uso requiere de la adquisición de una clave de API y su pago por uso. Leaflet hace uso del mapa del mundo OpenStreetMap, generado por la comunidad y de uso libre

Metodología

3.1 Metodología de desarrollo

Una metodología de desarrollo software (SDM, por sus siglas en inglés) es un *framework* utilizado para estructurar, planificar y controlar todo el proceso de desarrollo de un sistema de información. Garantiza una implementación adecuada del producto desde el momento de la idea hasta la fase de desarrollo.

En general, en este proyecto se ha intentado seguir enfoques ágiles, los cuales están basados en su gran mayoría en desarrollos iterativos promoviendo un proceso de desarrollo disciplinado que fomenta la inspección o adaptación frecuentes y valores como la autoorganización y la responsabilidad.

Debido a que el equipo de desarrollo está compuesto por una sola persona, se han seguido también algunos principios del Proceso de Software Personal (PSP) el cual propone a los ingenieros de software una forma disciplinada de estructurar su trabajo personal (Universidad de Granada, s.f.). Entre los objetivos de PSP destacan el planificar el trabajo, para ello en este caso se ha optado por utilizar el método Kanban, basado en un tablero en el que se representan todas las tareas pendientes. Las tareas, representadas comúnmente por “tarjetas”, se mueven a través de las diferentes etapas del trabajo hasta la finalización. Algunos otros objetivos son el esforzarse para cumplir la planificación y

esforzarse para obtener productos de mejor calidad. Para obtener productos de mejor calidad, PSP establece un proceso de mejora, este proceso queda reflejado en la ilustración 11.

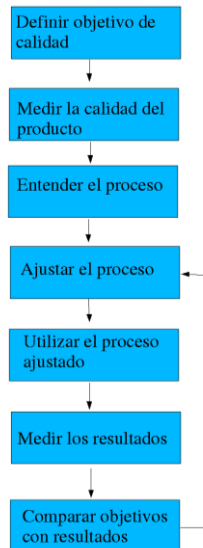


Ilustración 11 Proceso de mejora en PSP

En relación con el enfoque ágil, también se ha optado por usar el ciclo de vida ágil, uno de los fundamentos de estas metodologías. La principal característica de este ciclo es su carácter repetitivo, por ello también es conocido como un ciclo de vida iterativo o incremental. Este carácter repetitivo se da en etapas cortas y ágiles, donde se van añadiendo pequeñas funcionalidades al producto, siempre siendo testadas. Siendo muy similar a lo que ocurre en Scrum.

En la ilustración 12, se puede observar de forma visual en qué consiste cada una de las fases.



Ilustración 12 Ciclo de vida ágil

4

Análisis del sistema

4.1 Descripción

La fase de análisis define los requisitos del sistema, independientemente de cómo se cumplirán estos requisitos. Esta fase define y da forma al problema que el cliente está tratando resolver.

El resultado final de esta fase es un documento de requisitos, donde en este caso se incluyen requisitos funcionales, no funcionales, la especificación de los casos de uso y los diagramas de secuencia.

Normalmente, el documento de requisitos describe las “cosas” en el sistema y las “acciones” que pueden realizar estas cosas, pero no implican un diseño de arquitectura, sino una descripción de los artefactos del sistema y cuál es su comportamiento, desde la perspectiva del cliente. Más tarde, en la fase de diseño, estas descripciones de requisitos se asignan a distintas estructuras de datos y algoritmos.

4.2 Requisitos funcionales

Los requisitos funcionales de un sistema son aquellas funciones o servicios que se espera que el sistema suministre, cómo reaccionará a determinadas

entradas o cómo se comportará en situaciones particulares. Para este sistema se han identificado los siguientes requisitos funcionales (RF):

- RF1: Gestión de Usuarios
 - RF1.1. Registro. Los usuarios pueden registrarse en el sistema/plataforma.
 - RF1.2. Validar acceso. La aplicación debe ser capaz de permitir el paso a los usuarios registrados en la aplicación.
 - RF1.3. Cerrar sesión. Los usuarios pueden cerrar sesión el sistema/plataforma.
- RF2: Conexión con dispositivo de mediciones
 - RF2.1. Conexión a Arduino: Los usuarios pueden conectarse al módulo Bluetooth previamente emparejado mediante su dispositivo móvil.
 - RF2.2. Desconexión de Arduino: El usuario puede finalizar la conexión con el módulo bluetooth desde su dispositivo móvil.
- RF3: Comunicación con dispositivo de mediciones:
 - RF3.1. Recogida de datos: Los usuarios pueden recoger en su dispositivo móvil las mediciones de los sensores instalados en el Arduino mediante la comunicación bluetooth.
- RF4: Gestión de las mediciones
 - RF4.1. Subida de mediciones. El usuario desde su dispositivo móvil puede enviar las mediciones de los distintos sensores al servidor web.
 - RF4.2. Almacenamiento de las mediciones. El sistema es capaz de almacenar un histórico con las mediciones tomadas por los distintos

usuarios desde sus dispositivos móviles teniendo en cuenta la ubicación de donde fueron tomadas estas.

- RF4.3. Visualización de las mediciones. El usuario puede visualizar desde la aplicación móvil y desde la aplicación web un histórico de mediciones según la localización de donde fueron tomadas.

4.3 Requisitos no funcionales

Los requisitos no funcionales definen propiedades y otras restricciones del sistema, pero no establecen ninguna funcionalidad nueva. Para este sistema se han identificado los siguientes requisitos no funcionales (RNF):

- RNF1: Encriptación de los datos. Encriptación de los datos en tránsito con HTTPS y también de los datos en reposo.
- RNF2: Cumplir con el marco legal. Cumplimiento de la normativa del RGPD y la LOPDGDD.
- RNF3: Compatibilidad Android. La aplicación móvil debe ser compatible con dispositivos Android con una versión superior a la compilación 21 del SDK, es decir, Android 5.0 API (Lollipop).
- RNF3: Compatibilidad navegadores. El cliente web debe poder visualizarse y ser funcional en los principales navegadores (Chrome, Firefox, Microsoft Edge).
- RNF4: Diseño intuitivo: La aplicación móvil y el cliente web tendrán un diseño claro, sencillo, directo e intuitivo.

4.4 Casos de uso

Para definir los casos de uso, en primer lugar se va a utilizar el lenguaje UML para diseñar los diagramas de casos de uso y en segundo lugar, para definir la especificación de los casos del uso del sistema se va a utilizar la plantilla que propone Craig Larman en su libro Applying UML and Patterns (Larman, 2004).

4.4.1 Diagramas de casos de uso

En este caso se va a realizar una distinción entre el usuario que accede a la aplicación móvil para tomar las mediciones y el usuario que accede a la aplicación web a consultar mediciones.

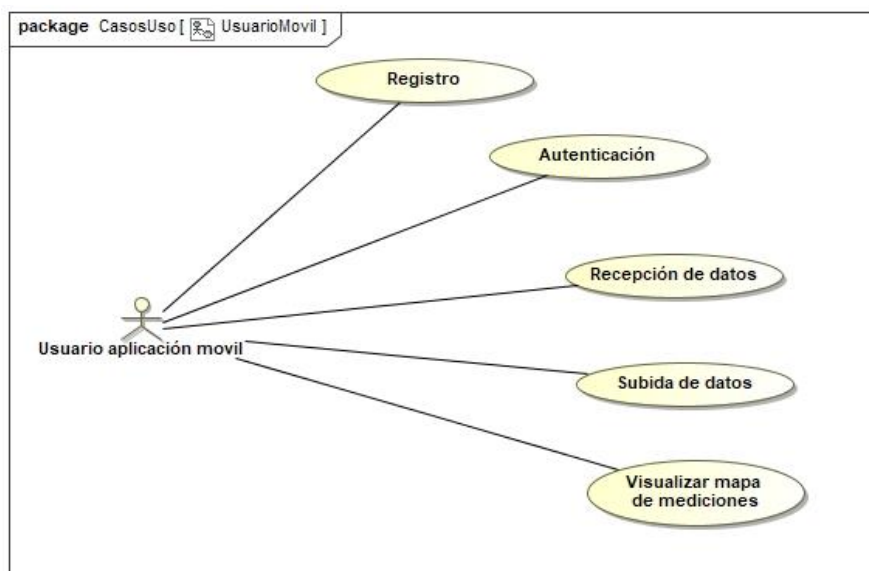


Ilustración 13 Casos de uso usuario móvil

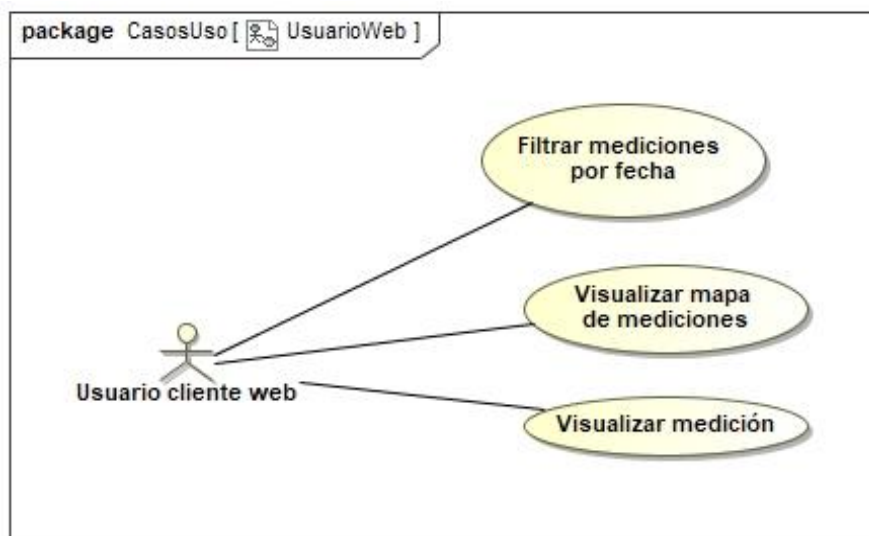


Ilustración 14 Casos de uso usuario web

4.4.2 Especificación casos de uso

| | |
|----------------------------|---|
| Caso de Uso | Registro |
| Alcance | WaterQuality Android App |
| Actor principal | Usuario dispositivo móvil |
| Precondiciones | Usuario sin registro previo |
| Garantía de éxito | - El usuario se encuentra conectado en la ventana “Inicio” |
| Escenario principal | <ol style="list-style-type: none"> 1. El usuario abre la aplicación en su dispositivo Android 2. El usuario presiona el botón Registrar 3. El usuario introduce el email en el campo “email” 4. El usuario introduce la contraseña en el campo “contraseña” 5. El usuario introduce la contraseña en el campo “repita contraseña” 6. El usuario presiona el botón “registrarse” |
| Extensiones | <p>6a. Se ha producido un error autenticando al usuario</p> <ol style="list-style-type: none"> 1. Se ha introducido un correo erróneo. Volver al paso 3 <p>6b. Las contraseñas no coinciden</p> <ol style="list-style-type: none"> 1. Las contraseñas de los campos “contraseña” y “repita contraseña” no coinciden Volver al paso 4 |

Tabla 2 CU Registro App

| | |
|------------------------|--|
| Caso de Uso | Autenticación |
| Alcance | WaterQuality Android App |
| Actor principal | Usuario dispositivo móvil |
| Precondiciones | Usuario con registro previo sin inicio de sesión |

| | |
|----------------------------|---|
| Garantía de éxito | - El usuario se encuentra conectado en la ventana "Inicio" |
| Escenario principal | <ol style="list-style-type: none"> 1. El usuario abre la aplicación en su dispositivo Android 2. El usuario introduce el email en el campo "email" 3. El usuario introduce la contraseña en el campo "contraseña" 4. El usuario presiona el botón "acceder" |
| Extensiones | <ol style="list-style-type: none"> 4a. Se ha producido un error autenticando al usuario <ol style="list-style-type: none"> 1. Se ha introducido un correo o contraseña erróneo. Volver al paso 3 |

Tabla 3 CU Autenticación App

| | |
|----------------------------|--|
| Caso de Uso | Recepción de Datos |
| Alcance | WaterQuality Android App |
| Actor principal | Usuario dispositivo móvil |
| Precondiciones | <ul style="list-style-type: none"> - Usuario con registro previo e inicio de sesión. - Usuario conectado en la ventana "Inicio" - Dispositivo móvil con Bluetooth activado - Dispositivo Arduino en servicio - Dispositivo Arduino vinculado con anterioridad |
| Garantía de éxito | - Se muestran los datos recibidos en pantalla |
| Escenario principal | <ol style="list-style-type: none"> 1. El usuario presiona el botón "Conectar Dispositivo" en la ventana de "Inicio" 2. El usuario selecciona el dispositivo Arduino en el listado 3. El usuario presiona el botón "Recibir datos" |
| Extensiones | <ol style="list-style-type: none"> 3a. La conexión falló <ol style="list-style-type: none"> 1. Volver al paso 2 y seleccionar el dispositivo Arduino |

Tabla 4 CU Recepción de datos App

| | |
|----------------------------|--|
| Caso de Uso | Subida de Datos |
| Alcance | WaterQuality Android App |
| Actor principal | Usuario dispositivo móvil |
| Precondiciones | <ul style="list-style-type: none"> - Usuario con registro previo e inicio de sesión - Usuario con datos a subir en pantalla |
| Garantía de éxito | <ul style="list-style-type: none"> - El usuario recibe un toast asegurando la subida de los datos - El usuario se encuentra conectado en la ventana “Inicio” |
| Escenario principal | 1. El usuario presiona el botón “Enviar datos” |

Tabla 5 CU Subida de datos App

| | |
|----------------------------|---|
| Caso de Uso | Visualizar mapa de mediciones |
| Alcance | WaterQuality Web |
| Actor principal | Usuario cliente web |
| Precondiciones | |
| Garantía de éxito | <ul style="list-style-type: none"> - El usuario visualiza un mapa con mediciones |
| Escenario principal | 1. El usuario accede a la dirección web |

Tabla 6 CU Visualizar mapa Web

| | |
|----------------------------|--|
| Caso de Uso | Filtrar mediciones por fecha |
| Alcance | WaterQuality Web |
| Actor principal | Usuario cliente web |
| Precondiciones | <ul style="list-style-type: none"> - El usuario visualiza un mapa con mediciones |
| Garantía de éxito | <ul style="list-style-type: none"> - El usuario visualizar un mapa que contiene únicamente las mediciones tomadas entre las fechas elegidas |
| Escenario principal | <ol style="list-style-type: none"> 1. El usuario introduce una fecha de inicio 2. El usuario introduce una fecha de fin 3. El usuario presiona el botón filtrar |

Tabla 7 CU Filtrar por fecha Web

| | |
|----------------------------|--|
| Caso de Uso | Visualizar medición |
| Alcance | WaterQuality Web |
| Actor principal | Usuario cliente web |
| Precondiciones | - El usuario visualiza un mapa que contiene mediciones |
| Garantía de éxito | - Se despliega un <i>pop-up</i> con la información de la medición |
| Escenario principal | 1. El usuario presiona sobre una “chincheta” en el mapa |
| Extensiones | 1a. Existe más de una medición para la misma ubicación 1. Volver al paso 1, sobre una de las “chinchetas” desplegadas |

Tabla 8 CU Visualizar medición Web

4.4.3 Diagramas de secuencia

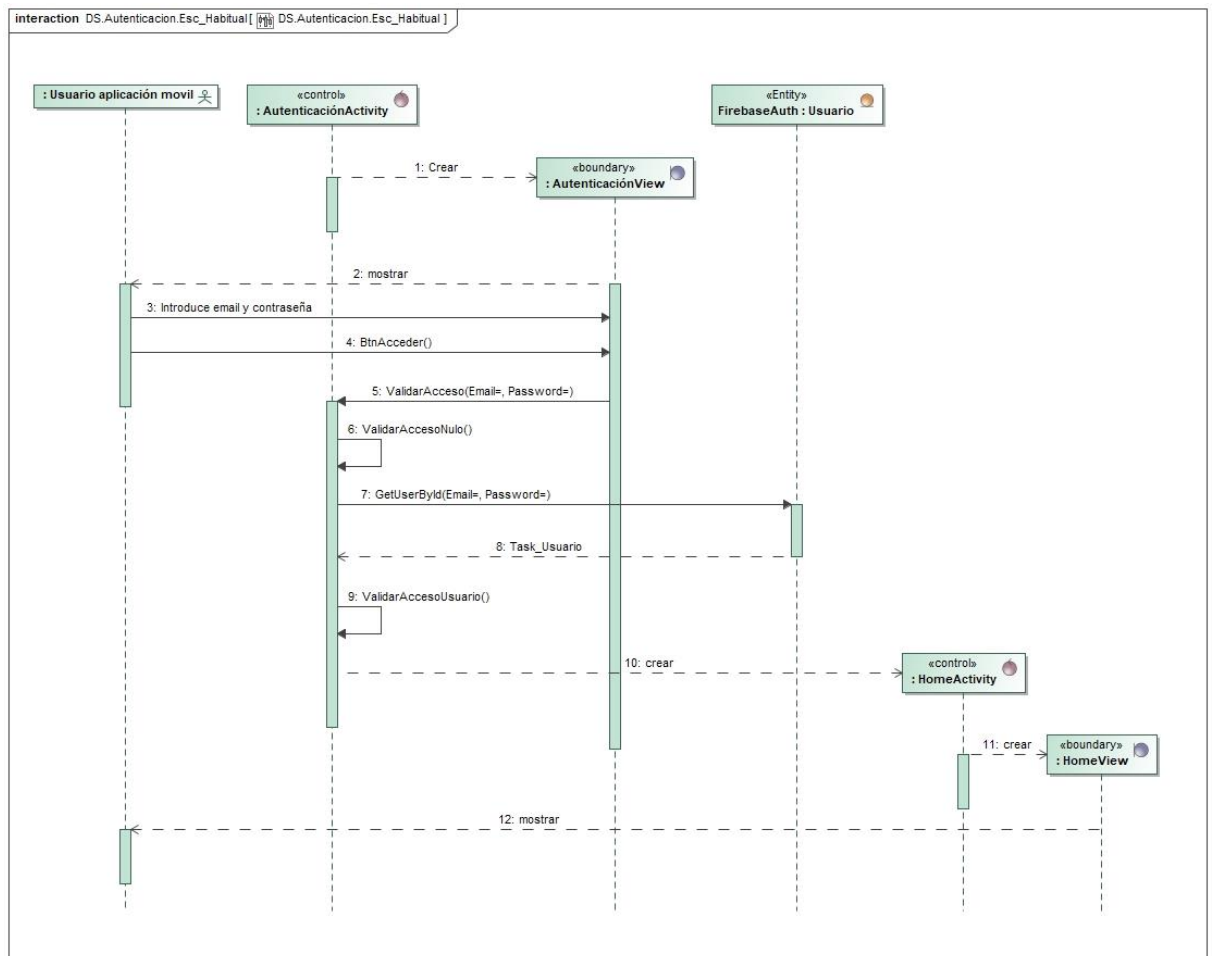


Ilustración 15 DS Autenticacion. EscHabitual

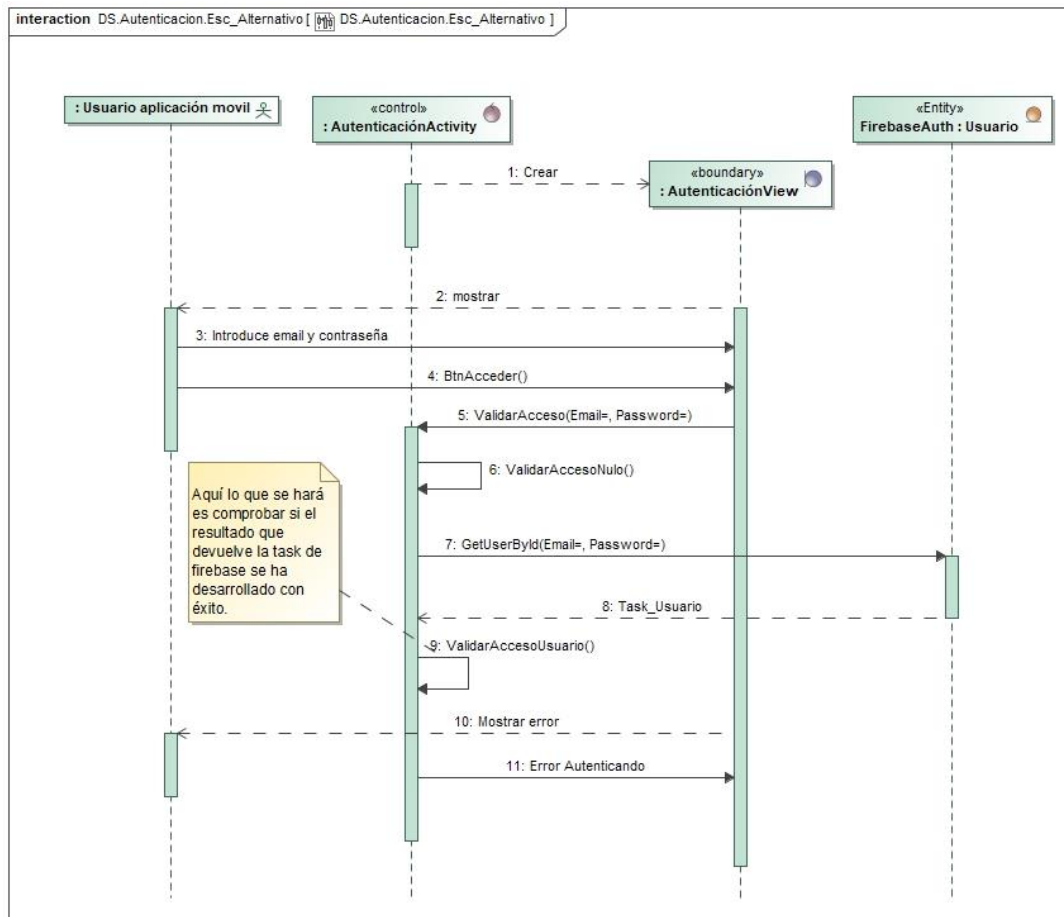


Ilustración 16 DS Autenticacion. EscAlternativo

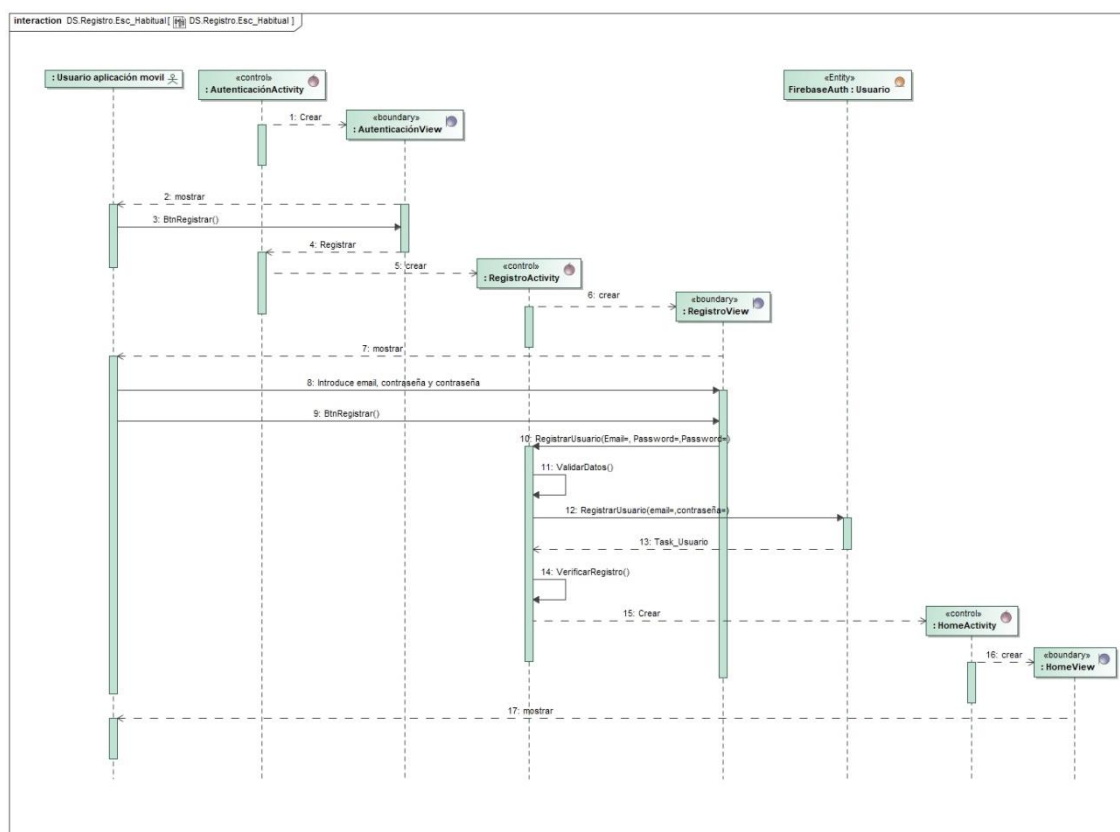


Ilustración 17 DS Registro. EscHabitual

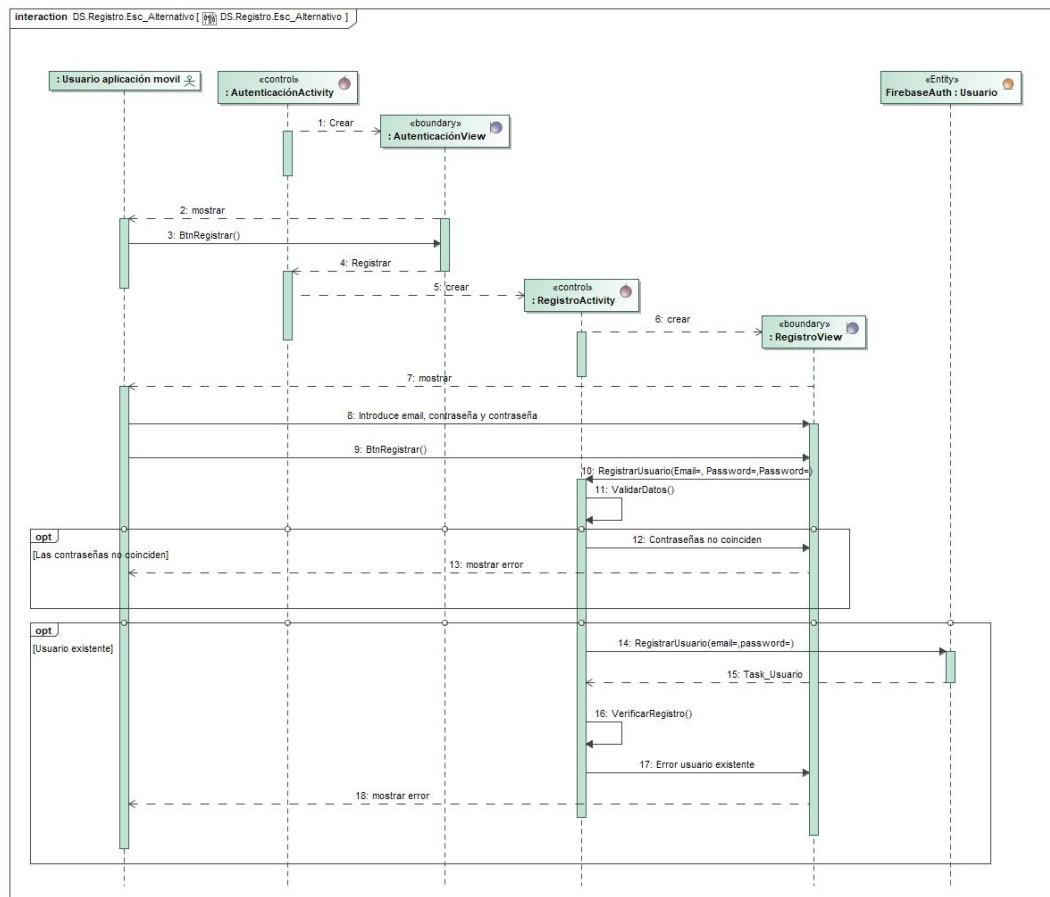


Ilustración 18 DS Registro. EscAlternativo

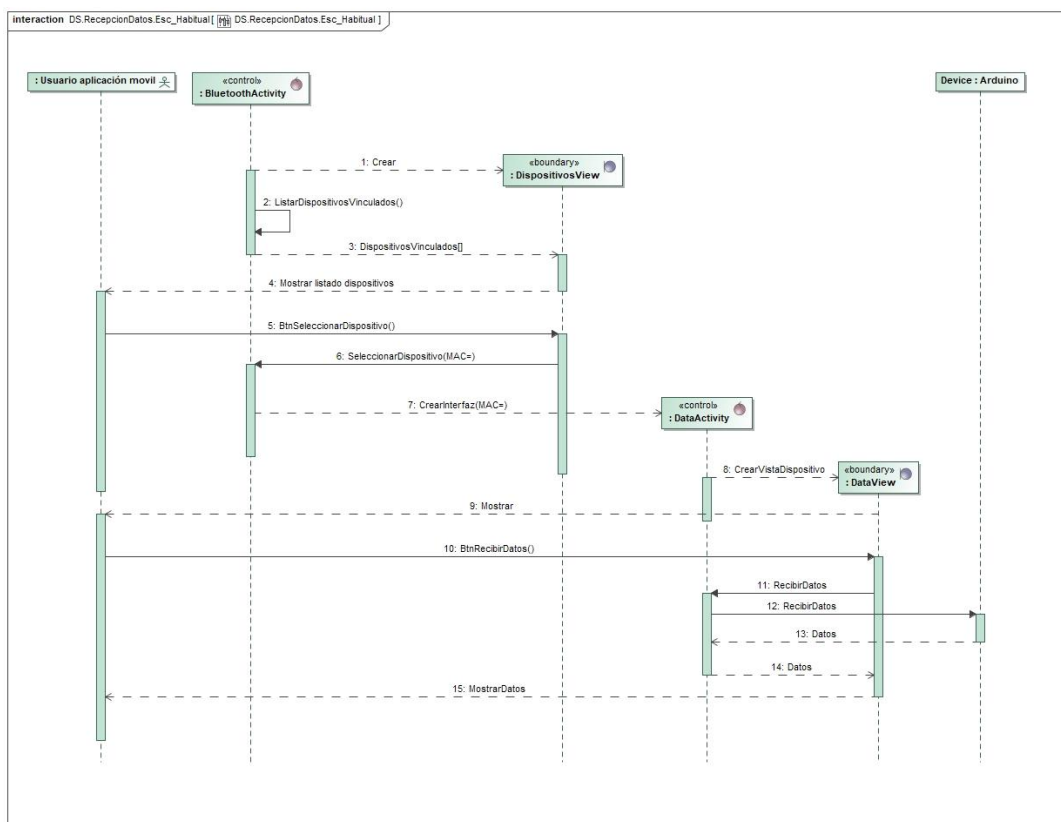


Ilustración 19 DS RecepcionDatos. EscHabitual

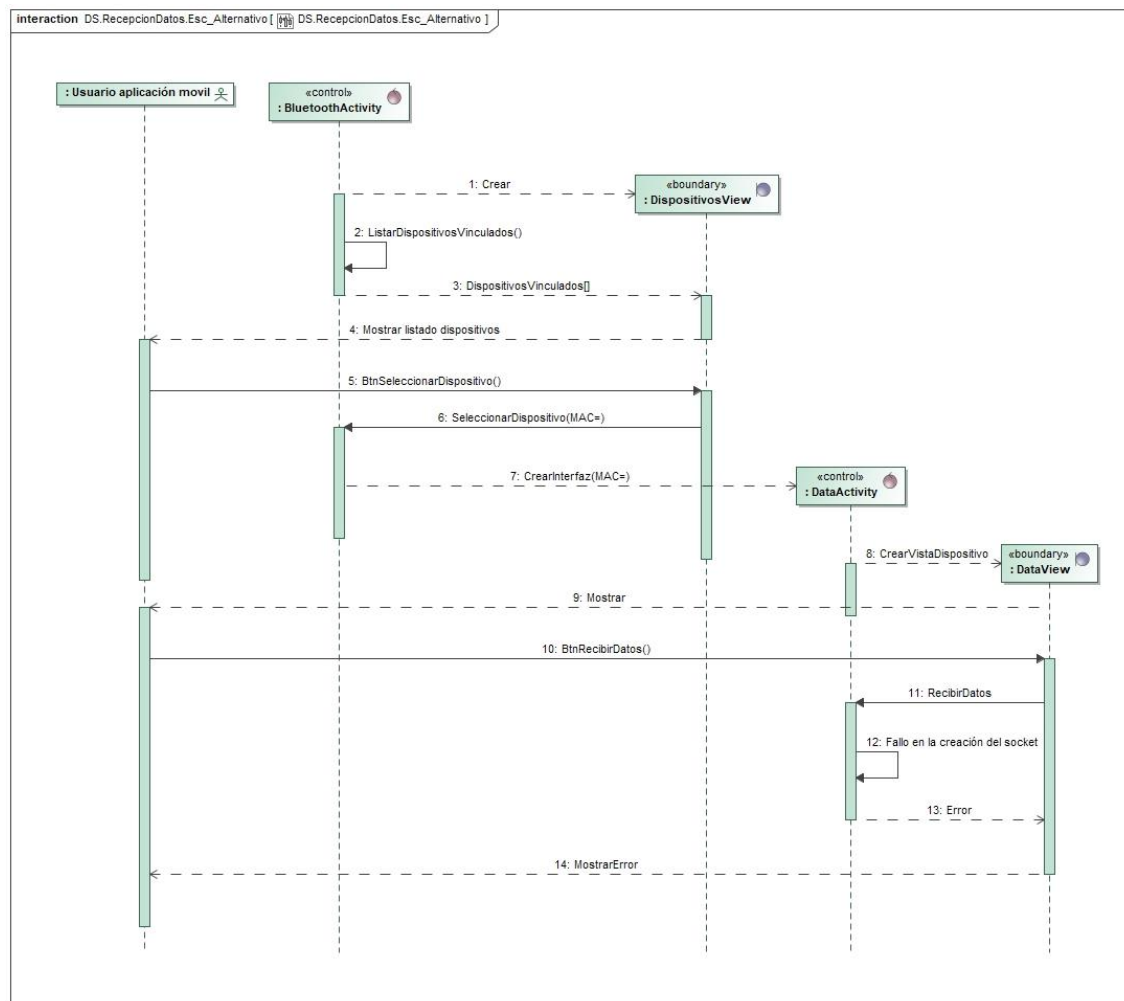


Ilustración 20 DS Recepción de Datos. EscAlternativo

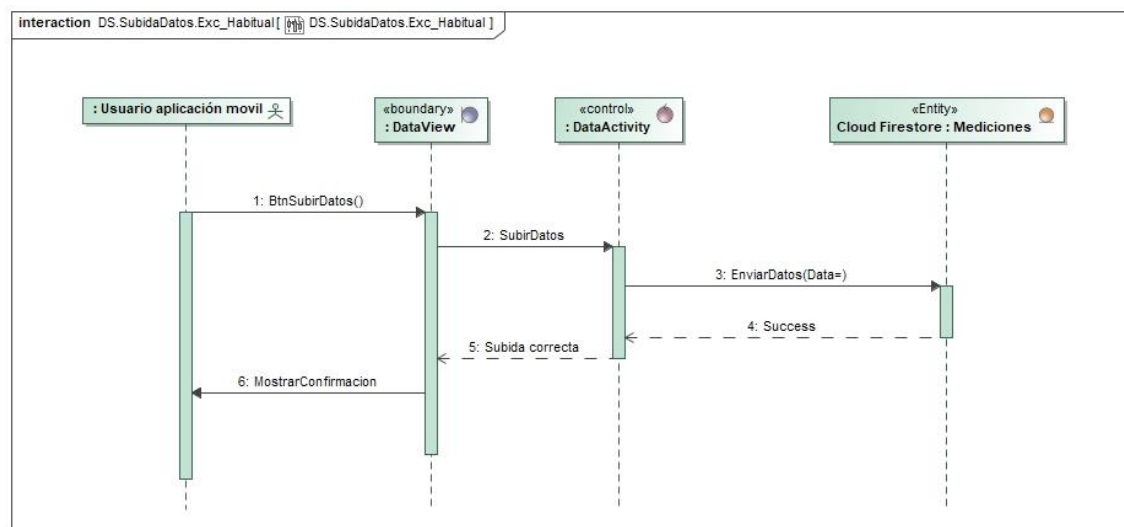


Ilustración 21 DS Subida de Datos. EscHabitual

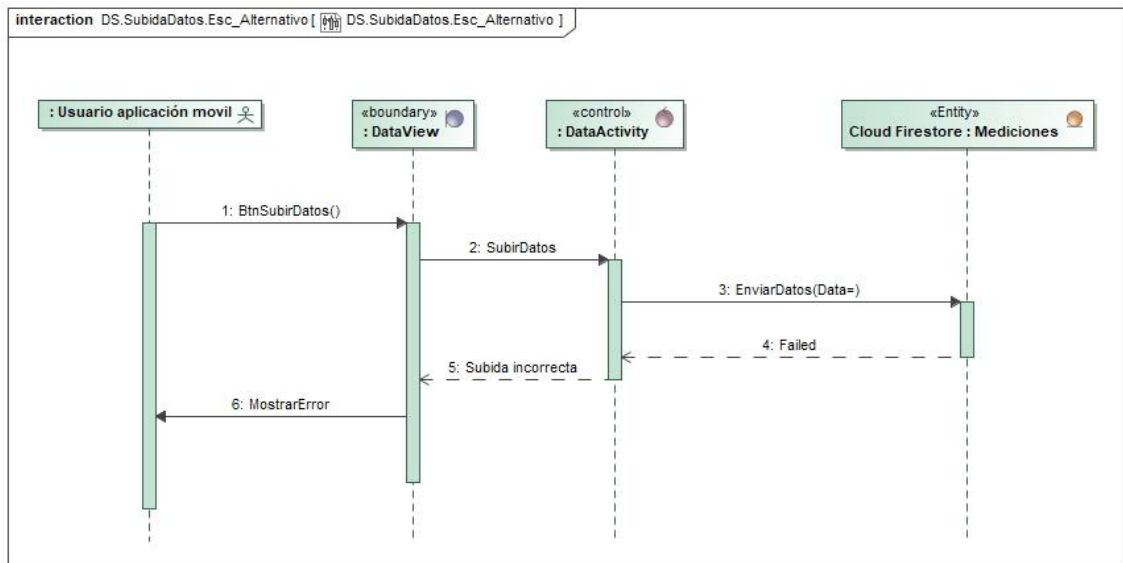


Ilustración 22 DS Subida de Datos. EscAlternativo

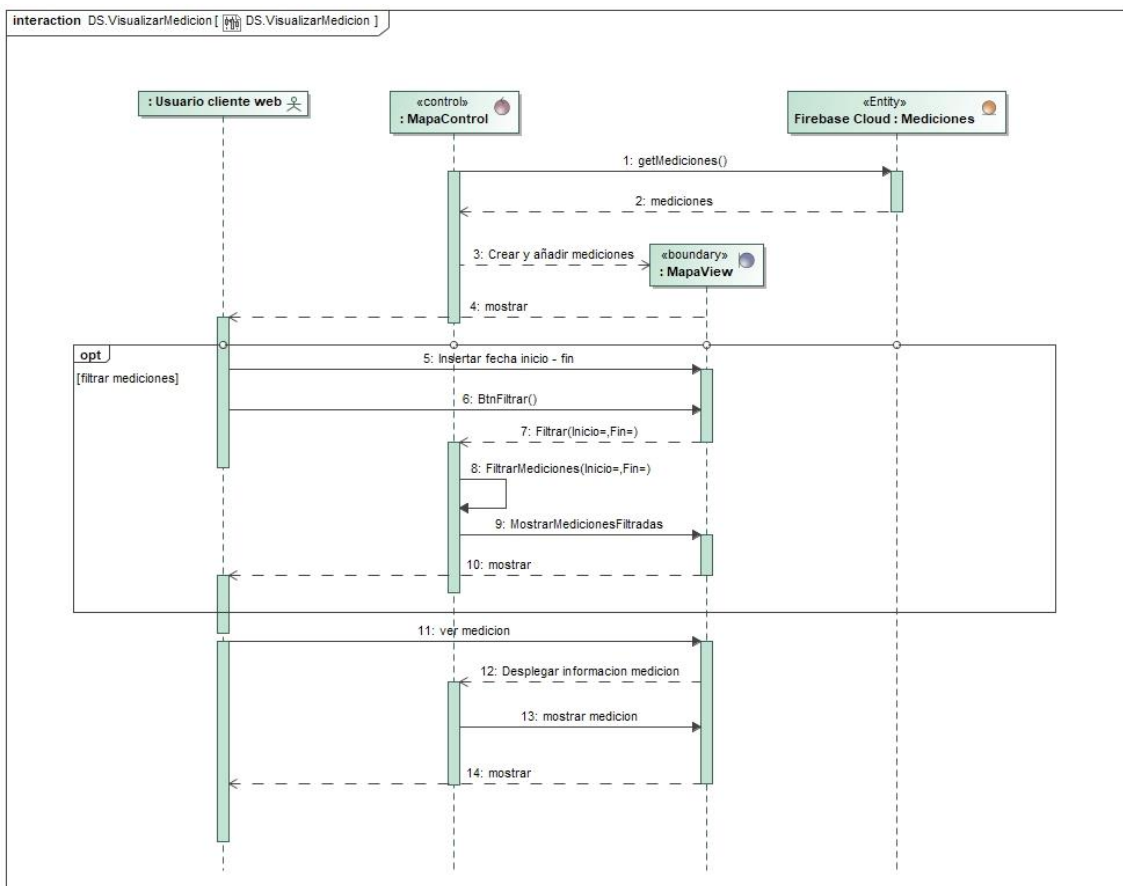


Ilustración 23 DS Visualizar medición

5

Diseño

5.1 Diseño de la aplicación móvil

Para el diseño de la aplicación móvil, en primer lugar se ha elaborado un diagrama de clases (Ilustración 24). Este diagrama tiene una estructura estática y representa la estructura del sistema mostrando sus clases, atributos, operaciones o métodos y las relaciones entre objetos.

Cada rectángulo representa una clase y el compartimento superior es el nombre de esta.

En el compartimento central se enumeran cada uno de los atributos de la clase en una línea separada. El atributo se encuentra dividido en dos partes, la parte anterior al signo de puntuación (:), que representa el nombre del atributo y la parte posterior que representa el tipo del atributo.

En el tercer compartimento se ubican las operaciones de la clase. Cada operación se encuentra dividido en dos partes, la parte anterior al signo de puntuación (:), que representa el nombre de la operación y la parte posterior que representa el tipo de valor retornado.

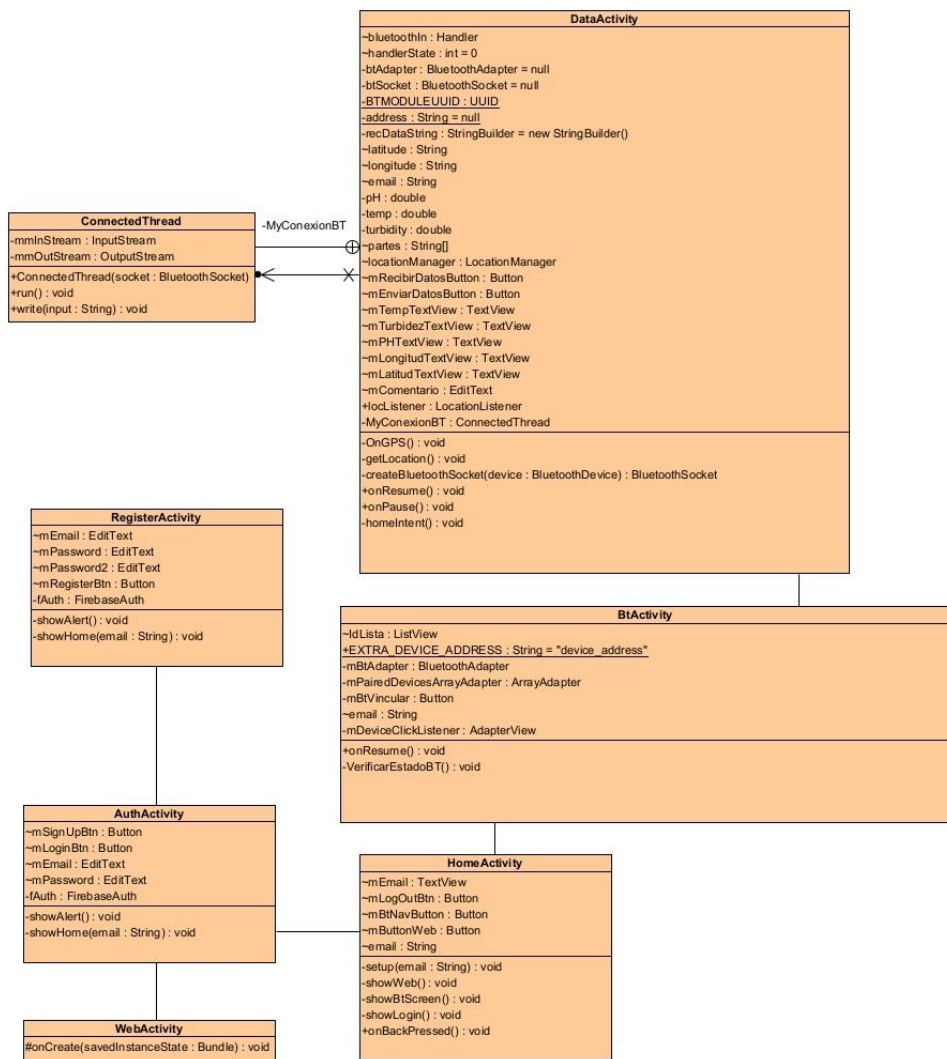


Ilustración 24 Diagrama de clases. Aplicación móvil

A continuación, se va a detallar el diseño para cada una de las partes de la aplicación.

5.1.1 Autenticación

La aplicación móvil mostrará inicialmente una pantalla de inicio de sesión donde se le dará la bienvenida al usuario a la aplicación.

Esta ventana contendrá dos cajas de texto para introducir el usuario, en este caso un email, y la contraseña del mismo. Finalmente, más abajo, el usuario tendrá un botón para poder acceder a la aplicación.

En caso de que el usuario no esté previamente registrado, tendrá un botón en la parte inferior de la misma pantalla para tal fin. A continuación, se explicará las características de la pantalla dedicada al registro.

5.1.2 Registro

En la pantalla para el registro, el usuario podrá observar en la parte superior que se encuentra en la pantalla destinada para tal fin, mejorando así la accesibilidad y el carácter intuitivo de la aplicación, como así lo requería uno de los requisitos no funcionales previstos.

El objetivo a lo largo de la aplicación es que los elementos condensados en una misma pantalla sean pocos, siempre con un aspecto grande, facilitando así el acceso a todos los públicos.

En la pantalla de registro, el usuario se encontrará con 3 cajas de introducción de texto. Cada una de estas cajas especificará claramente su fin. La primera será para la introducción del email, y las dos siguientes corresponderán a la introducción de la contraseña, siendo necesario repetirla para evitar problemas a la hora del registro, así el usuario podrá cerciorarse de que introduce la contraseña que está pensando.

Finalmente, el usuario dispondrá de un botón para finalizar el registro. En caso de que no sean iguales las contraseñas introducidas, el usuario podrá observar un mensaje que indica tal problema. Mientras que si ese email ya se encuentra registrado, también podrá observar un mensaje destinado a tal fin.

Tanto si el usuario ha accedido directamente desde la pantalla de inicio de sesión, como si ha accedido desde la pantalla de registro, será enviado a la pantalla de *home*, la cual es descrita a continuación.

5.1.3 Home

La pantalla de home será una pantalla donde el usuario, una vez se haya autenticado, disponga de un mensaje que le indique que se encuentra conectado, el email del usuario que se encuentra conectado y el botón principal será el de conectar dispositivo. Este botón guiará al usuario por una serie de pantallas hasta finalizar el proceso de envío de datos a la base de datos.

El usuario dispondrá de un botón de cierre de sesión, para desconectarse satisfactoriamente del usuario que previamente inició sesión.

Además, habrá disponible otro botón con el que el usuario podrá visualizar el mapa de mediciones sin necesidad de abrirlo en un navegador del dispositivo móvil o un ordenador.

5.1.4 Listado de dispositivos

Desde el la pantalla *home*, el usuario pulsa el botón conectar dispositivo, esto le lleva a una pantalla donde se podrán observar los dispositivos vinculados previamente en un listado. Es un requisito el que el dispositivo Arduino esté previamente vinculado al dispositivo móvil, ya que algunos módulos bluetooth compatibles con Arduino requieren de contraseña y esta configuración se realiza de forma más coherente y satisfactoria desde los propios ajustes del dispositivo Android. Es por esto que el usuario podrá observar en la parte inferior de la pantalla un mensaje indicándole que si no encuentra su dispositivo debe vincularlo con anterioridad. Para facilitar la accesibilidad y la operabilidad de la aplicación, se incluirá un botón debajo de este mensaje, mediante el cual el usuario podrá navegar directamente a los ajustes bluetooth de su dispositivo móvil Android desde la propia aplicación.

En el momento que el usuario pulse sobre uno de los dispositivos vinculados, se pasará a otra pantalla donde el usuario podrá recibir y enviar los datos.

5.1.5 Recepción y subida de datos

En esta pantalla, el usuario podrá observar una serie de parámetros que serán medidos a través del dispositivo Arduino. Inicialmente estos valores estarán a 0.

En la parte inferior de la pantalla, el usuario observará dos botones, uno que estará destinado a la recepción de los datos a través del dispositivo Arduino seleccionado previamente. El segundo botón estará destinado al envío de los datos que se muestran en pantalla a la base de datos para que posteriormente sean visualizados en la aplicación web. Además el usuario dispondrá de un campo de texto para añadir un comentario en relación a las sensaciones que detecte en el momento de la medición, fomentado así la actividad colaborativa del sistema.

Si el usuario no dispone de conexión a Internet en el momento de la subida, basta con pulsar el botón destinado al envío de los datos y salir de la aplicación, sin cerrar el proceso que la ejecuta. De este modo, una vez se recupere la conexión a Internet, todas las peticiones pendientes se subirán automáticamente a la nube.

5.1.6 Visualización del mapa

Esta vista estará destinada a mostrar el mapa de mediciones de la misma forma que se haría en la aplicación web y con las mismas funcionalidades

5.2 Diseño de la aplicación web

Al igual que se ha realizado con la aplicación móvil, para la aplicación web se ha diseñado un diagrama de clases (Ilustración 25).

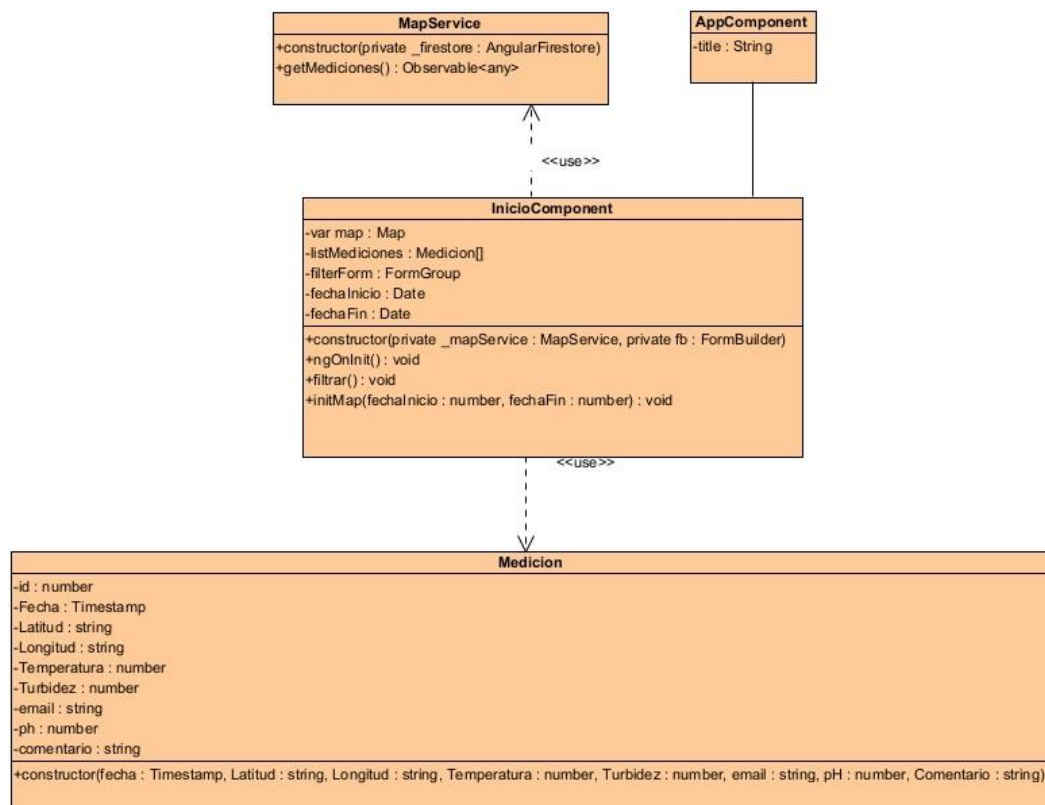


Ilustración 25 Diagrama de clases. Aplicación web

El objetivo de la aplicación web es que sea capaz de mostrar las mediciones tomadas previamente por otros usuarios desde la aplicación móvil.

Para ello, se dispone de un mapa que ocupa la mayor parte de la pantalla en el que mediante etiquetas se representa la presencia de una o varias mediciones en el lugar. Presionando cualquiera de estas etiquetas se podrá ver información detallada de la medición.

Además, una funcionalidad extra es que en la parte superior se dispondrá de una barra con el nombre de la aplicación y además dos cajas de texto y un botón de búsqueda. La utilidad de las cajas de búsqueda será introducir una fecha de inicio y una fecha de fin a la hora de filtrar las mediciones, ya que las mediciones se almacenan de forma histórica en las diferentes ubicaciones. Estas cajas de búsqueda permitirán el despliegue de un calendario para facilitar la elección de los días correspondientes y el botón realizará el filtrado correspondiente.

5.3 Diseño de la base de datos

Para el diseño de la base de datos se ha seguido el estilo que marca el servicio de Cloud Firestore. Esta base de datos es NoSQL. El hecho de que sea NoSQL aporta algunas características interesantes como la facilidad de organizar datos simples en documentos muy similares a JSON, e incluso la facilidad de almacenar datos complejos y jerárquicos en escala a través de subcolecciones..

El modelo de datos NoSQL de Cloud Firestore indica que “se almacenan los datos en documentos que contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones, que son contenedores para los documentos y que se pueden usar para organizar los datos y compilar consultas. Los documentos admiten varios tipos de datos diferentes, desde *strings* y números simples, hasta objetos anidados complejos”. (Google Firebase, s.f.). En concreto, los tipos de datos que admite almacenar Cloud Firestore son: vectores, booleanos, bytes, fecha y hora (*timestamp*), número de coma flotante, punto geográfico (desaconsejado, mejor almacenar latitud y longitud como numéricos), número entero, mapa (clave, valor), nulo, referencias a otros documentos o *strings* de texto. Es decir, la jerarquía sería la siguiente: **colección** → **documento** → datos

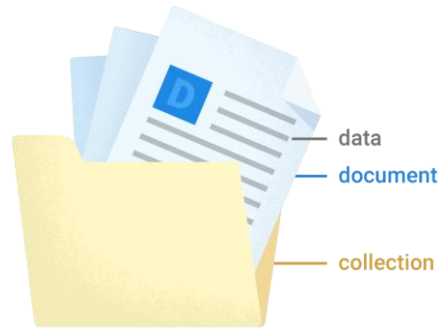


Ilustración 26 Modelo de datos NoSQL de Cloud Firestore

En este caso se tendrá una única colección, que será la de las mediciones. Dentro de esta colección se encuentran los documentos, que representan cada una de las mediciones de los distintos usuarios. Cada uno de estos documentos contendrán la siguiente información:

- Fecha: se trata de un valor con formato *timestamp* el cual almacenará la fecha y hora en la que se tomó la medición. Internamente este valor corresponde al número de milisegundos desde el 01/01/1970 00:00:00 UTC.
- Latitud: valor de tipo *string* que representa la latitud del lugar de la medición.
- Longitud: valor de tipo *string* que representa la longitud del lugar de la medición.
- Temperatura: valor de tipo numérico que nos aporta la temperatura del agua en el momento de la medición.
- Turbidez: valor de tipo numérico que nos aporta la turbidez del agua en el momento de la medición.
- pH: valor de tipo numérico que nos aporta la acidez o alcalinidad del agua en el momento de la medición.
- email: es el identificador del usuario de tipo *string* que envía la lectura, se almacena internamente de cara a tener identificadas las mediciones y evitar que estén anonimizadas, evitando así acciones fraudulentas en el caso de un mal uso de la aplicación.

- Comentario: campo de tipo *string* que almacena el comentario del usuario que realiza la medición.

6

Implementación

6.1 Implementación de la aplicación móvil

6.1.1 Autenticación

La pantalla que se despliega al abrir la aplicación es la de autenticación. Esta pantalla está compuesta por elementos *TextView* para mostrar los textos correspondientes al título y un mensaje para preguntar al usuario si no se ha registrado previamente. Para introducir el email y la contraseña se han utilizado elementos *EditText*, con la particularidad que para el email se ha elegido que este campo sea de tipo de entrada *textEmailAddress* y para la contraseña de tipo *textPassword*.

Tanto el botón de acceso como el de registro, son elementos de tipo *Button*.

Para iniciar sesión, en primer lugar obtenemos una instancia de *FirebaseAuth*, para ello es importante que previamente se haya registrado la aplicación Android en Firebase. Esto se consigue proporcionando a Firebase el nombre del paquete de nuestra aplicación y en ese momento, Firebase nos proporcionará un archivo llamado *google-services.json*, el cual se debe insertar en la raíz del proyecto. Este documento contiene toda la información relativa al proyecto como por ejemplo el *project_number*, *project_id* o el *api_key*.

Para activar el servicio de Authentication basta con presionar *Comenzar* dentro de Authentication en la consola de Firebase.

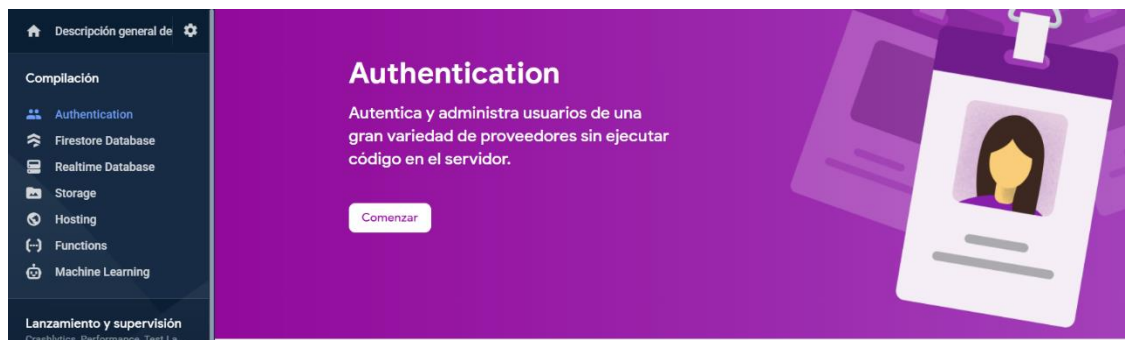


Ilustración 27 Servicio Authentication Firebase

Dentro del servicio de Authentication, se puede elegir el proveedor de acceso que tendrá la aplicación. En este caso va a ser mediante Correo electrónico/contraseña (Ilustración 28).

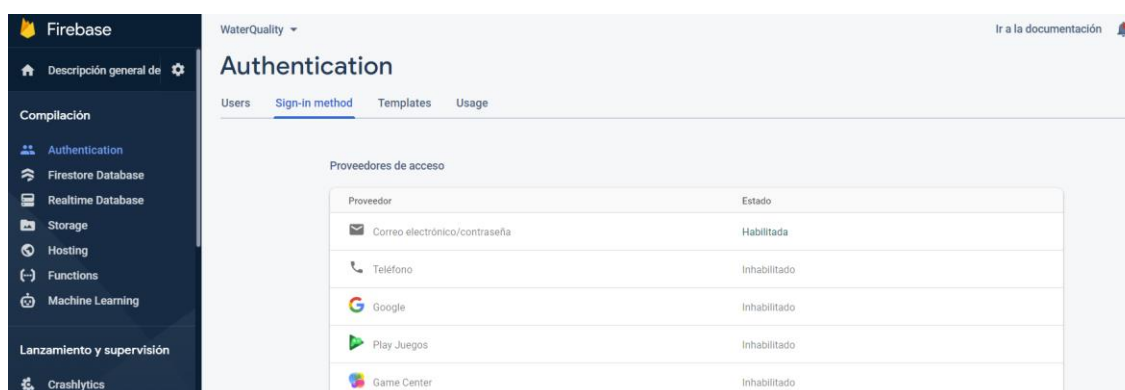


Ilustración 28 Proveedores de acceso. Authentication

En la pestaña *Users* se puede ver la información correspondiente a los usuarios que se han dado de alta en la aplicación, así como la posibilidad de reestablecer la contraseña, inhabilitar la cuenta o borrarla (Ilustración 29).

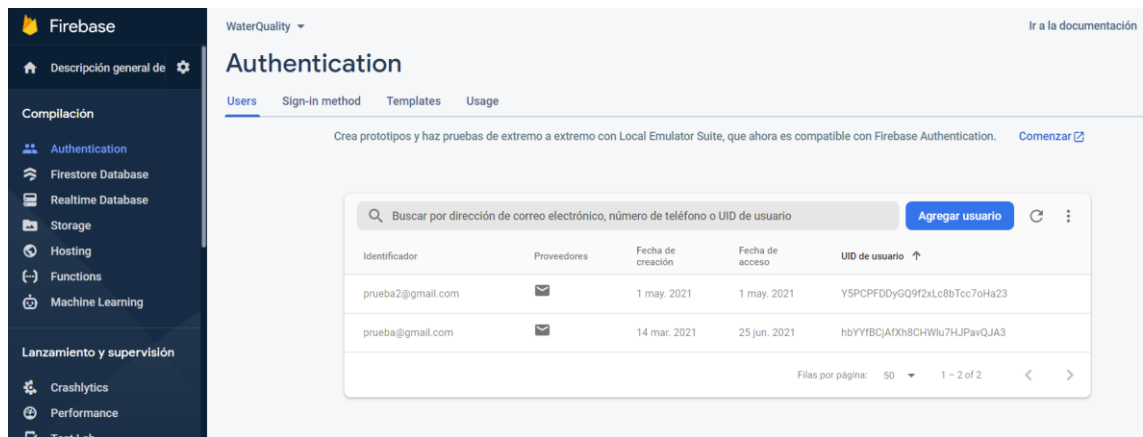


Ilustración 29 Gestión Usuarios. Authentication

Continuando con la implementación de la aplicación en Android Studio, cuando el usuario haya introducido los datos y pulse el botón de acceso, existirá un *listener* que se lanzará y efectuará la extracción de email y la contraseña de los *EditText* y si estos no son vacíos, llamará al método `signInWithEmailAndPassword(email, password)` de la instancia de `FirebaseAuth`. Si la respuesta es satisfactoria, se creará un *Intent* que redigirá al usuario a la pantalla de *home*, pasando un *Extra* que será el email del usuario. En caso contrario se creará un *AlertDialog* que mostrará un *pop-up* notificando de la existencia de un error autenticando al usuario.

6.1.2 Registro

Esta pantalla tendrá una estructura muy similar a la de autenticación. En este caso tendremos 3 elementos *EditText*, uno de ellos de tipo *textEmailAddress* y los otros dos de tipo *textPassword*.

El funcionamiento también es muy similar, en este caso se llama a la función `createUserWithEmailAndPassword(email, password)` que devuelve un `isSuccessful` en el caso de que haya ido todo bien y se creará un *Intent* que redigirá al usuario a la pantalla de *home*, pasando un *Extra* que será el email del usuario. Previo a esta llamada, se hace una comprobación de que las contraseñas introducidas en ambos *EditText* coinciden, en caso contrario y siempre que las contraseñas sean distintas, se indica un mensaje a tal fin.

6.1.3 Home

En esta pantalla un elemento principal es un *textView* que muestra el email del usuario que ha iniciado sesión, este email se extrae mediante un objeto de la clase *Bundle* a partir del Extra que se adjuntó en la creación del *Intent*.

Otro elemento es el botón para navegar hacia el listado de dispositivos Bluetooth, esto será un *Button* que creará un *Intent* hacia la actividad correspondiente. El *button* “mapa de mediciones” creará un *Intent* hacia la actividad *MapActivity*.

El elemento restante de esta pantalla es un *button* que realiza una llamada a *signOut()* a través de la instancia de *FirebaseAuth* y crea un *Intent* hacia la pantalla de autenticación.

6.1.4 Listado de dispositivos

En esta pantalla, el elemento principal es un *ListView* que mostrará todos los dispositivos que han sido vinculados en el teléfono Android. Para ello, la aplicación en primer lugar realiza una verificación de que el dispositivo soporte bluetooth y que este esté activado, todo esto gracias a la clase *BluetoothAdapter* y sus métodos *getDefaultAdapter()*, que devolverá null si el dispositivo no soporta bluetooth, y el método *isEnabled()*, que como su propio nombre indica, devolverá True si se encuentra activado. En caso de que no esté activado, se lanzará un *Intent* que simplemente muestra un mensaje en pantalla donde se indica que la aplicación desea activar el bluetooth y disponiendo de un botón para permitirlo, a este *Intent* se le pasa el valor *BluetoothAdapter.ACTION_REQUEST_ENABLE* para conseguir tal fin.

Posteriormente, para conseguir listar todos los dispositivos vinculados, se hace uso del método *getBondedDevices()* de la clase *BluetoothAdapter*, que devuelve un conjunto (Set) de tipo *BluetoothDevice*.

Finalmente, para poder pulsar en un elemento de la lista y que este sea seleccionado se ha configurado un *OnItemClickListener*, el cual se encarga dentro de su método *onClickItem* de crear un nuevo *Intent* hacia la pantalla de

recepción y envío de datos, pasando en el mismo la dirección MAC del dispositivo bluetooth elegido.

6.1.5 Recepción y subida de datos

En la pantalla de recepción y subida de datos se listan una serie de *textView* que mostrarán información acerca de los parámetros medidos una vez se pulse el botón de recibir datos. Además, se muestra un *editText* para introducir el comentario del usuario antes de realizar la subida de los datos.

Cuando se pulsa el botón de recibir datos, se intenta enviar un carácter específico al dispositivo elegido, esto se hace a través de la creación de un socket bluetooth con el dispositivo. En el caso de que la creación de este socket falle, es decir, no se produzca la conexión, se muestra un *Toast* informando de que la creación falló.

En ese momento, el dispositivo Arduino enviará el mensaje pertinente y mediante un Handler podremos procesar dicho mensaje. En el momento en el que se reciba el carácter “~”, se dará por hecho que se ha recibido el mensaje completo y se procederá a “trocear” el mensaje en los distintos parámetros. Además, en este paso se obtiene la ubicación del dispositivo, todo esto a través de la clase *LocationManager*. Aquí también se comprobará el escenario en el que la ubicación GPS no esté activada, pidiendo al usuario que la active y una vez sea así, se obtendrá la latitud y longitud correspondientes.

Cuando se pulse el *Button* enviar datos, se obtendrá la instancia correspondiente de *Firestore* y mediante un *Map<String, Object>*, se creará una estructura en la que el *String* hace referencia al atributo que aparecerá posteriormente en la base de datos, y el *Object* será el valor. Este *Map* será añadido a la Base de datos mediante el método *add()* de la instancia creada, donde también se le indica la colección de la base de datos a la que va destinada la estructura. Finalmente se mostrará un *Toast* en caso de que la subida de datos se haya completado con éxito y otro en caso contrario.

6.1.6 Visualización del mapa

En esta vista se visualizará un *WebView* a pantalla completa del mapa de mediciones. Este *WebView* cargará la página web donde se encuentra alojada la aplicación a través del método *loadUrl()*. Además, se obligará a utilizar la orientación horizontal del dispositivo móvil con el método *setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE)*; y se omitirá la *ActionBar* superior con el método *getSupportActionBar().hide()*; para conseguir una correcta y cómoda visualización del mapa.

6.2 Implementación de la aplicación web

La aplicación Angular es desarrollada utilizando la Angular CLI (Command Line Interface), de esta forma se crea el proyecto y los servicios y componentes correspondientes.

Para la implementación de la aplicación web se ha elegido crear un proyecto Angular donde se hace uso de Bootstrap para la creación de la interfaz gráfica. Como la aplicación no requiere una cantidad excesiva de cómputo en los datos, Firestore es la única tecnología de *backend* usada, donde el *frontend* extrae los datos requeridos como se explicará más adelante.

La aplicación web está compuesta por un componente en el que predomina un mapa. Para este mapa se ha utilizado la librería *leaflet* y la funcionalidad *Leaflet.markercluster* para crear despleables (Ilustración 30) en el caso de que haya más de una medición en un mismo lugar.



Ilustración 30 Leaflet MarkerCluster

Al constructor de la clase se le pasa el servicio *MapService*, el cual se ha implementado para que se encargue de crear la conexión con la base de datos mediante el propio SDK de Firebase Javascript. Para usar esto es necesario

añadir una variable en el *environment.ts* y en *environment.prod.ts*, esta variable tendrá valores tales como la *apiKey* o el *projectId*.

Mediante el componente Inicio, recibiremos las mediciones a través del método *subscribe()* de la clase *Observable*, el patrón de observador es un patrón de diseño de software en el que un objeto, llamado sujeto, mantiene una lista de sus dependientes, llamados observadores, y les notifica automáticamente los cambios de estado.

Una vez se tienen los datos, se genera el mapa a través del método *map()* de Leaflet, se crean los diferentes *markers*, que son las mediciones, en función de si se encuentran dentro de la fecha indicada por el usuario o no. En el caso de que el usuario haya introducido estas fechas, para extraerlas se hace uso de la clase FormBuilder, implementando la vista del componente que muestra el formulario reactivo incrustado en un *navbar* de Bootstrap en la misma vista y añadiendo en el constructor del componente un componente *FormGroup* que contendrá un campo *fromDate* y otro *toDate*. En el método *filtrar()* que se lanza cuando se pulsa el *button* de tipo *submit* se guardan en variables los valores obtenidos desde el formulario de la vista, y se llama al método *initMap(fechaInicio, fechaFin)*.

6.3 Implementación de la base de datos

La base de datos utilizada en este sistema será Cloud Firestore. Para la creación de esta base de datos es necesario tener previamente creado un proyecto de Firebase, tal y como ocurre en este caso. Simplemente es necesario pulsar *Crear base de datos* en la pestaña Cloud Firestore de la consola del proyecto (Ilustración 31).



Ilustración 31 Cloud Firestore

A continuación, se pregunta si se desea iniciar en modo producción o en modo de prueba, siendo esta última la elegida, ya que permite habilita todo de tipo de operaciones durante 30 días para así poder realizar un proceso de desarrollo, siendo las reglas de seguridad predeterminadas del modo de prueba las siguientes:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2021, 6, 7);
    }
  }
}
```

Para configurar nuestro entorno de desarrollo Android, se utilizará la BoM de Firebase para Android previamente añadida al proyecto y declarando en este caso la dependencia hacia el producto Cloud Firebase, todo esto en el archivo Gradle a nivel de app:

```
// Declare the dependency for the Cloud Firestore library
// When using the BoM, you don't specify versions in Firebase
library dependencies
    implementation 'com.google.firebase:firebase-firestore'
```

Para configurar el entorno de desarrollo Web, se debe instalar la librería oficial de Firebase para Angular, esto se hace mediante:

```
ng add @angular/fire
```

6.4 Implementación de Arduino

6.4.1 Conexiones hardware

Para la implementación del dispositivo Arduino, en primer lugar se va a explicar las conexiones de los distintos componentes hardware:

Tanto el módulo del sensor de pH, el módulo del sensor de turbidez y el módulo del sensor de temperatura disponen de 3 conexiones:

- VCC. Es el pin que suministra potencia al módulo, debe ir conectado a una conexión VCC de 5V de la placa Arduino.
- GND. Es el pin necesario para generar polaridad en el circuito, más conocido como tierra o masa, va conectado al pin GND de la placa Arduino.
- ASO (Analog Signal Output). Es la conexión que se encarga de la transmisión de los datos hacia el dispositivo Arduino. La recepción de la información en la placa se produce gracias a que esta contiene un convertidor de analógico a digital multicanal de 10 bits. La conversión se realiza asignando voltajes de entrada entre 0 y 5V a valores enteros entre 0 y 1023.

En cambio, el módulo bluetooth se dispone de 6 conexiones:

- VCC. Con la misma funcionalidad que antes, es el pin que suministra potencia al módulo, debe ir conectado a una conexión VCC de 5V de la placa Arduino.
- GND. Es el pin necesario para generar polaridad en el circuito, más conocido como tierra o masa, va conectado al pin GND de la placa Arduino.
- RX. Es el pin de recepción del módulo, es decir, se encarga de recibir información del dispositivo Arduino, por lo tanto se conecta a uno de los pines de transmisión serie de la placa Arduino, es decir, TX.
- TX. Es el pin de transmisión del módulo, es decir, se encarga de enviar información al dispositivo Arduino, por lo tanto se conecta a uno de los pines de recepción serie de la placa Arduino, es decir, TX.
- STATE. Es un pin de estado, devuelve 1 cuando el módulo está conectado. En este caso no se va a utilizar este pin
- EN. Es el pin que se utiliza para entrar en el modo de configuración del módulo bluetooth, como por ejemplo para cambiar entre el modo esclavo o maestro. Este pin tampoco se utilizará en este caso.

En la ilustración 32, se puede observar cómo quedarían todos los componentes conectados.

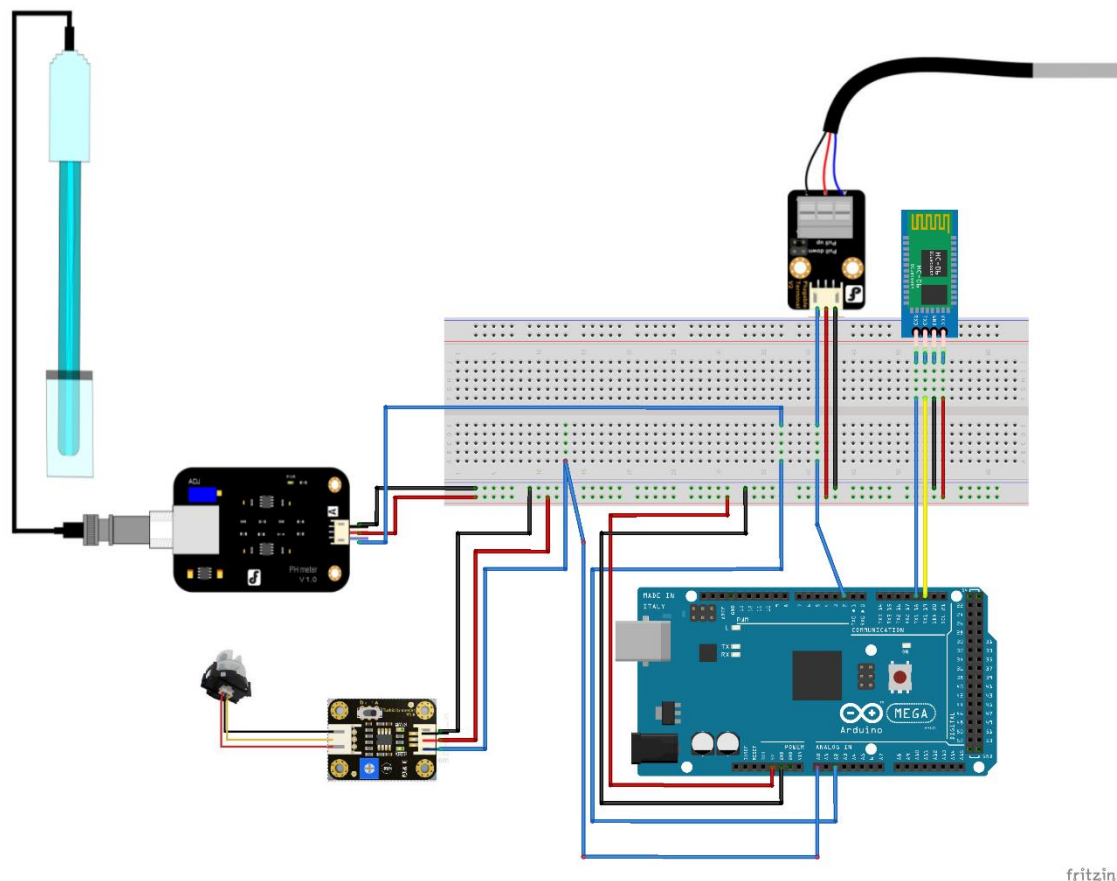


Ilustración 32 Diagrama de conexiones Arduino

6.4.2 Implementación software

Todos los sketch de Arduino siguen la misma estructura, la cual es algo peculiar y sencilla. La estructura se compone de tres partes esenciales:

```
//Comentarios introductorios
//Declaración de variables
//Importación de librerías

void setup() {
    // put your setup code here, to run once
    // Este método se ejecuta al inicio del programa, una sola vez
    // Normalmente, aquí se inicializan variables,
    //modos de pin, comienzo de uso de bibliotecas, etc.
}

void loop() {
    // put your main code here, to run repeatedly:
    // Este método, como su propio nombre indica, se repite
    //consecutivamente y permite que el programa cambie y responda
}
```

Para la obtención de los parámetros de pH y turbidez, se realizan mediciones a través de la función *analogRead()*, mediante la cual obtenemos un valor entre

0 y 1023. En función de lo proporcionado por el fabricante, se realizan las conversiones pertinentes para obtener el valor de la medición. En el caso de la turbidez se ha hecho un ensayo mediante 3 soluciones (Ilustración 32) con distinta turbidez para obtener una calibración aproximada.



Ilustración 33 Soluciones de calibración turbidez

Se ha elaborado la gráfica correspondiente para la conversión Volts-NTU a través de los puntos obtenidos (Ilustración 34).

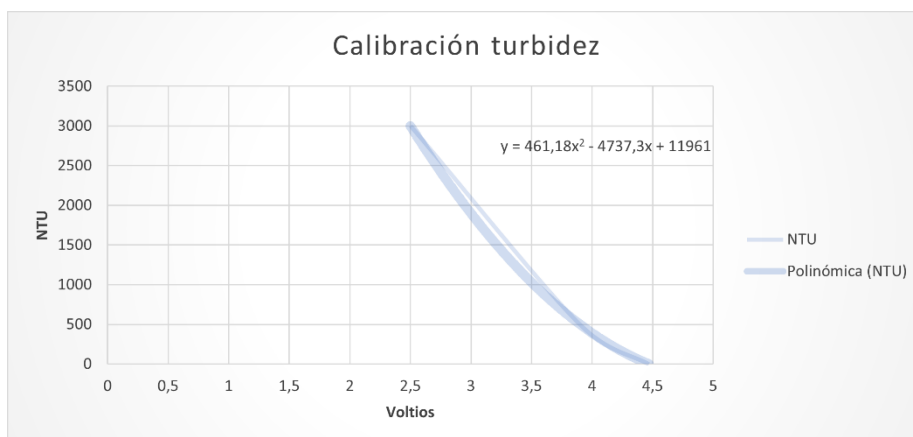


Ilustración 34 Gráfica Voltios-NTU

Para la calibración del sensor de pH serían necesarios una soluciones específicas de calibración (Ilustración 35), en este caso no ha sido posible adquirirlas por lo que se ha hecho uso de sustancias conocidas ácidas como el limón y su valor aproximado de pH o sustancias alcalinas como la lejía.



Ilustración 35 Soluciones de calibración

Para la obtención de la temperatura, se hace uso de la librería *OneWire* y mediante el comando *OneWire ds(DS18S20_Pin)*; le indicamos el pin digital en el que se encuentra conectado el sensor. El código utilizado detecta si hay más de un sensor en el mismo pin y presenta la temperatura de todos ellos, en este caso solo tenemos uno pero existiría la posibilidad de aumentar la cantidad

7

Pruebas

7.1 Introducción a la fase de pruebas

Las pruebas son una parte fundamental de cualquier proyecto de desarrollo software, normalmente se elabora un informe de calidad en el que se incluyen todos los puntos implicados en la correcta consecución de los requisitos de calidad que harán posible elevar el beneficio aportado por el sistema, evitando en gran medida fallos o déficits posteriores que limiten el éxito del producto.

Tanto en el desarrollo como al finalizar un proyectos, es importante someter a los sistemas desarrollados a pruebas. Estas pruebas dan, tanto al desarrollador como al cliente, las certezas necesarias para asegurar que el nivel de fallos y errores posteriores sea inexistente o muy limitado.

7.2 Pruebas realizadas

| Número de prueba | TST1 |
|------------------|---|
| Caso de prueba | El dispositivo Arduino realiza mediciones de forma coherente en función de la situación. Comprobando la salida de la consola del Arduino IDE, se muestra una temperatura elevada en el caso del agua caliente y una temperatura baja |

| | |
|------------------------------|---|
| | para un agua refrigerada. Para un agua turbia a simple vista, la turbidez es elevada y para un agua cristalina lo contrario. Para una sustancia ácida el valor mostrado se encuentra en valores ácidos, para una sustancia alcalina se muestran valores alcalinos y para una sustancia neutra se muestra un valor neutro. |
| Requisitos implicados | RF5.1. |
| Resultado | Correcto |

Tabla 9 Caso de prueba 1

| | |
|------------------------------|---|
| Número de prueba | TST2 |
| Caso de prueba | El dispositivo móvil se conecta con el dispositivo Arduino y los datos recibidos son iguales a los mostrados por la consola de Arduino. En caso de que la conexión no sea posible se muestra un error |
| Requisitos implicados | RF2.1. RF3.1. |
| Resultado | Correcto |

Tabla 10 Caso de prueba 2

| | |
|------------------------------|--|
| Número de prueba | TST3 |
| Caso de prueba | La aplicación móvil sube los datos recogidos del dispositivo Arduino a la base de datos, y los datos enviados y almacenados, teniendo en cuenta la ubicación, coinciden. Si la subida se realiza de forma incorrecta se muestra el mensaje correspondiente de error. |
| Requisitos implicados | RF4.1. RF4.2. |
| Resultado | Correcto |

Tabla 11 Caso de prueba 3

| | |
|-------------------------|---|
| Número de prueba | TST4 |
| Caso de prueba | El dispositivo Android finaliza el socket de conexión con el dispositivo Arduino una vez se sale de pantalla de transmisión de datos. |

| | |
|------------------------------|----------|
| Requisitos implicados | RF2.2. |
| Resultado | Correcto |

Tabla 12 Caso de prueba 4

| | |
|------------------------------|---|
| Número de prueba | TST5 |
| Caso de prueba | La aplicación móvil permite el inicio de sesión a usuarios registrados previamente. Si el usuario no está registrado se muestra un mensaje de error |
| Requisitos implicados | RF1.2. |
| Resultado | Correcto |

Tabla 13 Caso de prueba 5

| | |
|------------------------------|---|
| Número de prueba | TST6 |
| Caso de prueba | La aplicación móvil permite cerrar la sesión a usuarios autenticados previamente. |
| Requisitos implicados | RF1.3. |
| Resultado | Correcto |

Tabla 14 Caso de prueba 6

| | |
|------------------------------|---|
| Número de prueba | TST7 |
| Caso de prueba | La aplicación móvil permite el registro a usuarios cuyo correo no existe previamente. En caso de un problema en el registro se muestra el correspondiente mensaje de error. |
| Requisitos implicados | RF1.1. |
| Resultado | Correcto |

Tabla 15 Caso de prueba 7

| | |
|------------------------------|--|
| Número de prueba | TST8 |
| Caso de prueba | La aplicación web permite la visualización de las mediciones en un mapa y al pulsar sobre una medición o un conjunto de ellas se puede observar los datos de la misma. |
| Requisitos implicados | RF4.3. |
| Resultado | Correcto |

Tabla 16 Caso de prueba 8

| | |
|-------------------------|------|
| Número de prueba | TST9 |
|-------------------------|------|

| | |
|------------------------------|--|
| Caso de prueba | La aplicación web permite la visualización filtradas por fechas de las mediciones en un mapa |
| Requisitos implicados | RF4.3. |
| Resultado | Correcto |

Tabla 17 Caso de prueba 9

7.3 Firebase Test Lab

Firebase Test Lab es un servicio proporcionado por Firebase, el cual permite realizar pruebas automáticas en dispositivos seleccionables dentro de un listado en el que se incluyen más de 100 dispositivos diferentes. La prueba es tan sencilla como subir el fichero .APK de la aplicación y Test Lab realiza el examen correspondiente. En este caso, la prueba Robo analiza la estructura de la UI de la app y la explora periódicamente simulando las actividades de un usuario de forma automática.

Si Test Lab encontrase algún error, mostraría detalladamente donde ha sucedido y facilitaría el proceso de resolución.

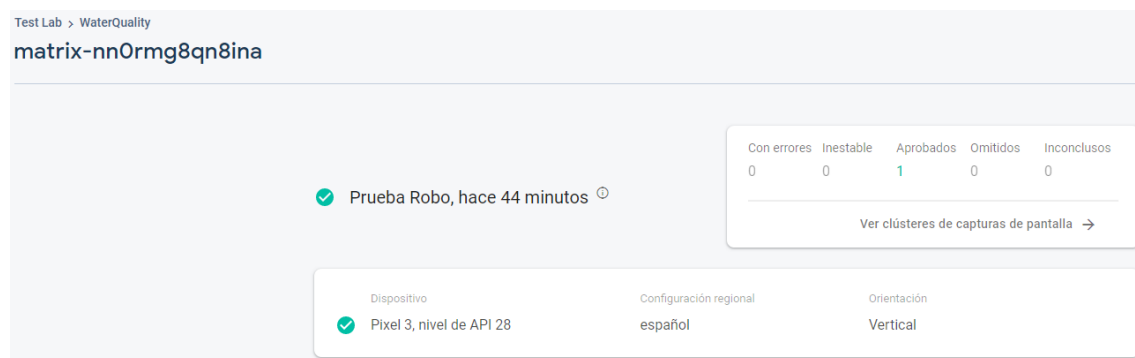


Ilustración 36 Prueba Test Lab

7.4 Firebase Crashlytics

Firebase proporciona una herramienta capaz de proveer informes de error en tiempo real. De forma integrada, cuando el usuario provoque algún fallo en la app, esta excepción es lanzada a la consola de Firebase de la aplicación y el desarrollador obtiene los datos suficientes, como por ejemplo la línea de código que lanza la excepción, para resolver este problema.

Esta es sin duda una de las formas más importantes de encontrar fallos en un desarrollo software, ya que el usuario final es realmente quien realiza las pruebas definitivas sobre un producto que normalmente siempre necesita estar actualizando y parcheando defectos.

Para añadir esta herramienta a la app es necesario habilitar Crashlytics y agregar el complemento a la app, esto desde el archivo `build.gradle` a nivel de proyecto mediante:

```
// Add the Crashlytics Gradle plugin (be sure to add version
// 2.0.0 or later if you built your app with Android Studio 4.1).
classpath 'com.google.firebase:firebase-crashlytics-gradle:2.6.1'
```

Y en el archivo `build.gradle` a nivel de app:

```
// Apply the Crashlytics Gradle plugin
apply plugin: 'com.google.firebase.crashlytics'
// Declare the dependencies for the Crashlytics and Analytics
libraries
// When using the BoM, you don't specify versions in Firebase
library dependencies
implementation 'com.google.firebase:firebase-crashlytics'
implementation 'com.google.firebase:firebase-analytics'
```

Finalmente, lo adecuado es forzar un fallo y es por ello que desde la documentación de Firebase nos proveen del siguiente código para efectuarla:

```
Button crashButton = new Button(this);
crashButton.setText("Crash!");
crashButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        throw new RuntimeException("Test Crash"); // Force a crash
    }
});

addContentView(crashButton, new ViewGroup.LayoutParams(
    ViewGroup.LayoutParams.MATCH_PARENT,
    ViewGroup.LayoutParams.WRAP_CONTENT));
```

De esta forma obtenemos información detallada en la consola del proyecto del error, asegurando que funciona de forma correcta (Ilustración 36).

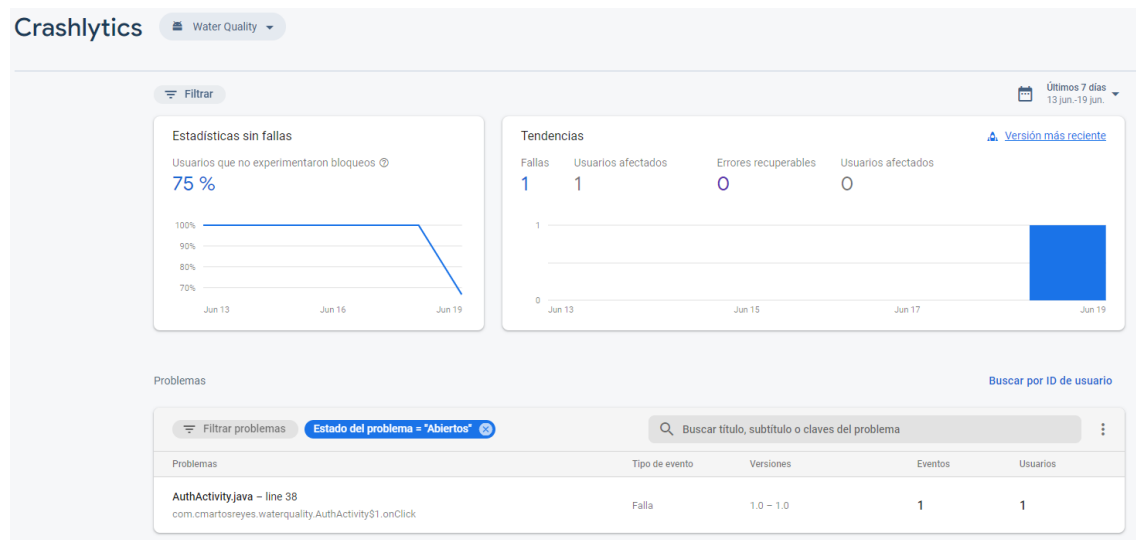


Ilustración 37 Consola de Crashlytics

8

Conclusiones y líneas futuras

8.1 Conclusiones

El desarrollo de este proyecto ha demostrado que el Internet de las Cosas (IoT) permite el desarrollo de sistemas capaces de resolver problemas de la sociedad o simplemente proveer de herramientas que son especialmente útiles para un gran abanico de usuarios.

Cada día se dispone de un mayor número de soluciones o servicios que permiten el desarrollo de este tipo de proyectos, y lo más importante es que se disponen de alternativas de código abierto al alcance de todos, las cuales se han priorizado en el desarrollo de este proyecto.

Otro punto a destacar es que a través de este proyecto se han pasado por todas las fases que componen un proyecto software. En primer lugar se estudió la necesidad existente y posteriormente se realizó una investigación de los distintos parámetros del agua a modo de determinación del ámbito del proyecto tal y como se haría en la fase de planificación de cualquier proyecto software. Estos conocimientos evidentemente quedan fuera del conocimiento de un

ingeniero informático, es por ello muy importante la capacidad analítica que se desarrolla durante el grado. Posteriormente se han pasado por las fases de análisis, diseño, implementación o pruebas, quedando muy claro la importancia de respetar cada una de estas fases. Además se ha hecho uso tanto de componentes hardware, siendo aquí Arduino y sus sensores los protagonistas, como de componentes software, pasando por desarrollos web y móvil. Completando así un gran abanico de tecnologías y herramientas.

Otros de los aspectos más importantes que se puede extraer sobre este proyecto es que pese al desconocimiento en ciertas tecnologías por parte de un desarrollador, la documentación que proporcionan estas es de vital ayuda para familiarizarse y aplicarlas de forma óptima, es por ello que lo que se extrae de aquí es que una tecnología lo suficientemente bien documentada tiene mayor posibilidad de éxito.

Tras todo lo anterior se concluye que se ha cumplido el principal objetivo del proyecto, el cual era crear un sistema basado en el Internet de las Cosas interconectando diferentes tecnologías y comunicaciones y que aportase la capacidad de poner a disposición de cualquier usuario un dispositivo económico y portátil para medir la calidad del agua.

8.2 Líneas futuras

Hay diferentes líneas futuras del sistema o simplemente mejoras que tendrían cabida, algunas de ellas serían:

- Representación de los datos. Añadiendo la posibilidad de representar la información mediante estadísticas o gráficas que proporcionen un punto de vista diferente de la información.
- Mayor cantidad de sensores. Añadiendo otros tipos de sensores, como por ejemplo, los de conductividad que pese a encontrarse por un precio algo más elevado, pueden seguir siendo asequibles para determinados públicos. Además, el sistema diseñado es perfectamente escalable en cuanto a cantidad de sensores por lo que no supondría mayor problema.

- Soluciones profesionales de calibración. Consiguiendo soluciones profesionales de calibración, y quizás sensores profesionales, el sistema se convertiría en un sistema profesional de medición de estos parámetros, estando todo bajo la misma arquitectura.
- Impresión 3D. Para facilitar el transporte del dispositivo se podría diseñar una caja de almacenamiento mediante impresión 3D para que las partes no resistentes al agua queden fuera de peligro.

Referencias

- Analizar mi agua. (19 de Agosto de 2019). *¿Cuáles son los parámetros de calidad del agua potable?* Obtenido de analizarmiagua.com:
<https://analizarmiagua.com/2019/08/16/parametros-de-calidad-del-agua-potable/>
- Arduino. (s.f.). *Introduccion*. Obtenido de
<https://www.arduino.cc/en/Guide/Introduction>
- Boletín Oficial del Estado. (21 de Febrero de 2003). *Real Decreto 140/2003, de 7 de febrero, por el que se establecen los criterios sanitarios de la calidad del agua de consumo humano*. Obtenido de
<https://www.boe.es/buscar/doc.php?id=BOE-A-2003-3596>
- Facsa. (23 de enero de 2017). *La dureza del agua*. Obtenido de
<https://www.facsa.com/la-dureza-del-agua/>
- Fondriest Environmental, Inc. (3 de Marzo de 2014). *"Conductivity, Salinity and Total Dissolved Solids." Fundamentals of Environmental Measurements*. Obtenido de
<https://www.fondriest.com/environmental-measurements/parameters/water-quality/conductivity-salinity-tds/>
- Global Water a xylem brand. (s.f.). *CAN YOU DETERMINE WATER HARDNESS FROM CONDUCTIVITY OR TOTAL DISSOLVED SOLIDS MEASUREMENTS?* Obtenido de
<http://www.globalw.com/support/hardness.html>
- Google Firebase. (s.f.). *Documentación Cloud Firestore*. Obtenido de
<https://firebase.google.com/docs/firestore?hl=es>
- Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. (s.f.). *Análisis de los resultados y conclusiones*. Obtenido de
http://ficus.pntic.mec.es/ngom0007/analisis_aguas.html
- Larman, C. (2004). *Applying UML and Patterns*. Pearson.
- Lenntech. (s.f.). *Turbidez*. Obtenido de <https://www.lenntech.es/turbidez.htm>
- Omer, N. H. (16 de octubre de 2019). *Water Quality Parameters*. Obtenido de intechopen.com: <https://www.intechopen.com/books/water-quality-science-assessments-and-policy/water-quality-parameters>
- Pradillo, B. (12 de septiembre de 2016). *Parámetros de control del agua potable*. Obtenido de iagua.es: <https://www.iagua.es/blogs/beatriz-pradillo/parametros-control-agua-potable>
- Stanford University. (s.f.). *The Analysis Phase*. Obtenido de Stanford University Infolab: <http://infolab.stanford.edu/~burback/watersluice/node4.html>
- United States Geological Survey. (s.f.). *pH and Water*. Obtenido de https://www.usgs.gov/special-topic/water-science-school/science/ph-and-water?qt-science_center_objects=0#qt-science_center_objects
- Universidad Complutense de Madrid. (14 de febrero de 2015). *Oxígeno Disuelto*. Obtenido de <https://www.ucm.es/data/cont/docs/952-2015-02-14-Oxigeno%20disuelto%20f.pdf>
- Universidad de Granada. (s.f.). *Proceso Software Personal*. Obtenido de <https://lsi2.ugr.es/~mvega/docis/psp.html>

YSI a xylem brand. (s.f.). *Turbidity Units, TSS, Water Clarity, Suspended Particles Measurement, Turbidity in Water*. Obtenido de <https://www.ysi.com/parameters/turbidity>

Apéndice A

Manual de Instalación

Requerimientos

- Arduino IDE
- Dispositivo Android con una versión del sistema operativo Android 5 o superior, ya que se trata de la versión mínima soportada por la aplicación móvil
- Visual Studio Code
- Nodejs y Angular

Instalación de Arduino

Para la instalación de la placa Arduino se detallaran cómo instalar los diferentes módulos que conforman el dispositivo.

Para el módulo de pH:

- Conectar el pin VCC a un pin de 5V de la placa Arduino
- Conectar el pin GND a un pin de GND de la placa Arduino
- Conectar el pin ASO al pin Analógico A2 de la placa Arduino

Para el módulo de temperatura:

- Conectar el pin VCC del módulo a un pin de 5V de la placa Arduino
- Conectar el pin GND del módulo a un pin de GND de la placa Arduino
- Conectar el pin ASO del módulo al pin PWM 0 de la placa Arduino

Para el módulo de turbidez:

- Conectar el pin VCC del módulo a un pin de 5V de la placa Arduino
- Conectar el pin GND del módulo a un pin de GND de la placa Arduino
- Conectar el pin ASO del módulo al pin Analógico A0 de la placa Arduino

Para el módulo de bluetooth:

- Conectar el pin VCC del módulo a un pin de 5V de la placa Arduino
- Conectar el pin GND del módulo a un pin de GND de la placa Arduino
- Conectar el pin RX del módulo al pin de comunicaciones 18 TX1 de la placa Arduino
- Conectar el pin TX del módulo al pin de comunicaciones 19 RX1 de la placa Arduino

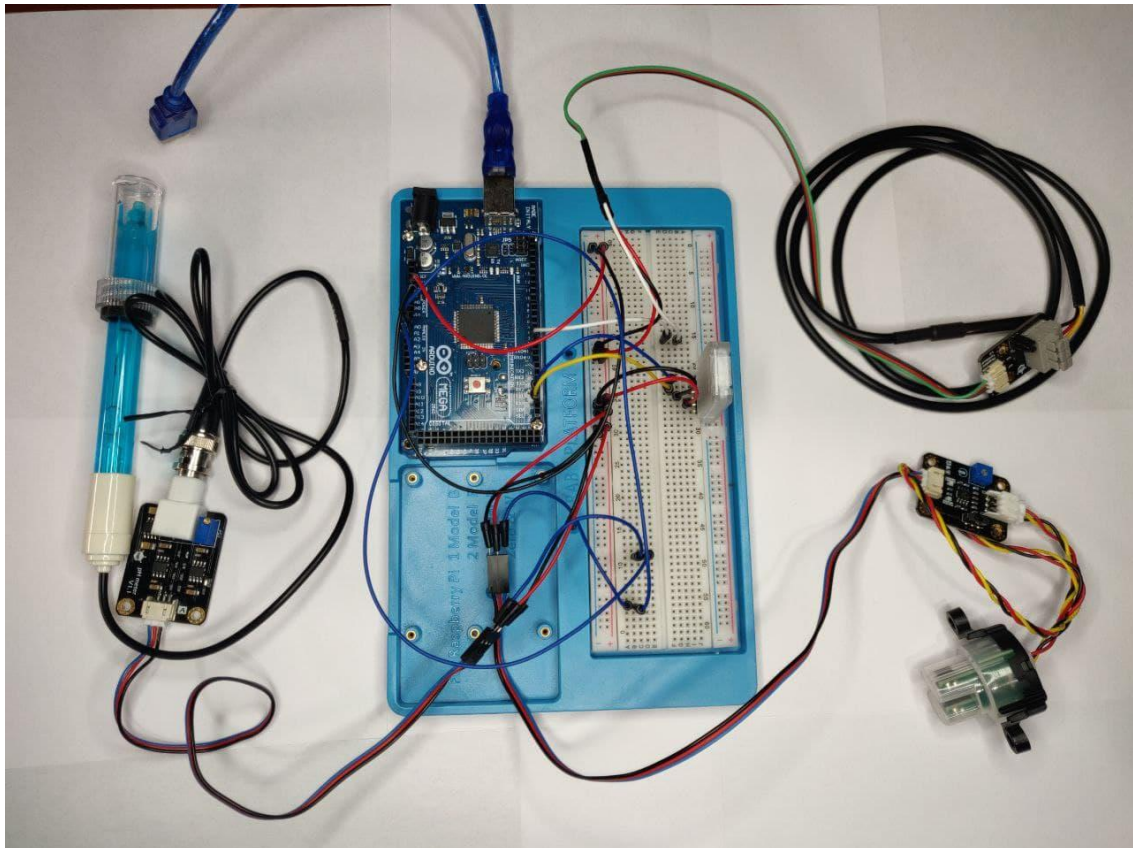


Ilustración 38 Instalación Arduino

Instalación de aplicación móvil

1. Permitir la instalación de aplicaciones de orígenes desconocidos

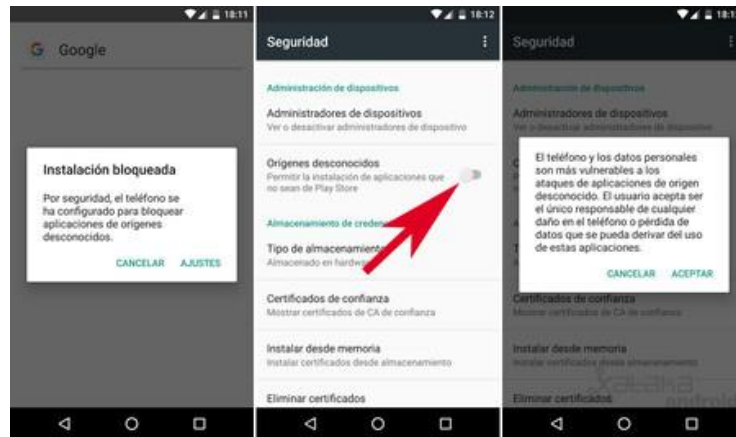


Ilustración 39 Instalación Aplicaciones Desconocidas

2. Descargar la aplicación del repositorio
<https://github.com/cmartosreyes/WaterQuality/blob/master/app/release/WaterQualityApp.apk>
3. Instalar la aplicación desde el explorador de archivos del dispositivo

Instalación de aplicación web

1. Abrir la carpeta del proyecto con Visual Studio Code. Este se puede importar desde el repositorio ubicado en
<https://github.com/cmartosreyes/WaterQualityWeb>
2. Abrir la consola dentro del entorno y ejecutar `npm install`
3. Utilizar el comando `ng serve -o`
4. La aplicación web corre en <http://localhost:4200/>
5. La aplicación se encuentra también desplegada en Internet, basta con buscar la dirección <https://waterquality-tfg.web.app/>

Apéndice B

Manuales de Usuario

Manual de Arduino

1. Abrir sketch_main en Arduino IDE. Este se puede descargar desde https://github.com/cmartosreyes/WaterQuality-Arduino/blob/main/sketch_main/sketch_main.ino
2. Conectar dispositivo Arduino a un pc a través de USB
3. En la barra superior de Arduino IDE seleccionar Herramientas → Placa → Arduino Mega 2560
4. Pulsar el “tick” para verificar y compilar el código
5. Pulsar la flecha hacia la derecha para subir el código al Arduino

Manual de la Aplicación móvil

Para hacer uso de esta aplicación es imprescindible haber seguido los pasos descritos en “Apéndice A – Instalación de Arduino”, “Apéndice A – Instalación de Aplicación móvil” y “Apéndice B – Manual de Arduino”

Una vez se hayan completado satisfactoriamente estos pasos, se puede proceder a abrir la aplicación.

Añadir medición

Para añadir una nueva medición se deben seguir:

Creación de cuenta

Este paso se debe realizar si es la primera vez que se accede a la aplicación, en caso contrario se puede pasar al apartado “Autenticación”

Para poder crear una cuenta es necesario pulsar el botón “Registrar” en la pantalla de autenticación (Ilustración 40).

WaterQuality

Bienvenido a WaterQuality

Email

Contraseña

ACCEDER

¿Aún no estás registrado?

REGISTRAR

Ilustración 40 Pantalla Inicio. Botón registro

En esta pantalla de registro se debe introducir un email en el campo destinado para tal efecto y una contraseña que por protección debe ser introducida dos veces (Ilustración 41, paso 1).

Finalmente se debe pulsar el botón “Registrarse” (Ilustración 41, paso 2) y si todo ha ido bien, debe aparecer una pantalla “Home” donde se indica el correo del usuario registrado (Ilustración 43).

Registro

Regístrese Aquí

1

Email

Contraseña

Repita Contraseña

REGISTRARSE 2

Ilustración 41 Pantalla Registro

Autenticación

Esta es la pantalla que aparece siempre que se abre la aplicación.

Para proceder con la autenticación, es necesario introducir el email previamente registrado y la contraseña (Ilustración 42, paso 1). Finalmente se deberá pulsar el botón “Acceder” (Ilustración 42, paso 2) y si todo ha ido bien debe aparecer una pantalla “Home” donde se indica el correo (Ilustración 43).

11:33 83%

WaterQuality

Bienvenido a WaterQuality

Email

Contraseña 1

ACCEDER 2

¿Aún no estás registrado?

REGISTRAR

Ilustración 42 Pantalla de autenticación

Vincular/Conectar Dispositivo

Para realizar esta acción, es necesario que se pulse el botón “Conectar Dispositivo” de la pantalla de Inicio (Ilustración 43).



Ilustración 43 Pantalla de Inicio

De esta forma se llega a la pantalla de dispositivos vinculados, aquí se debe pulsar el nombre del módulo bluetooth instalado en el dispositivo Arduino. En caso de que el dispositivo no aparezca en la lista, es el momento de vincularlo, para ello hay que pulsar el botón “Vincular” que directamente desplazará al usuario a los ajustes de Bluetooth

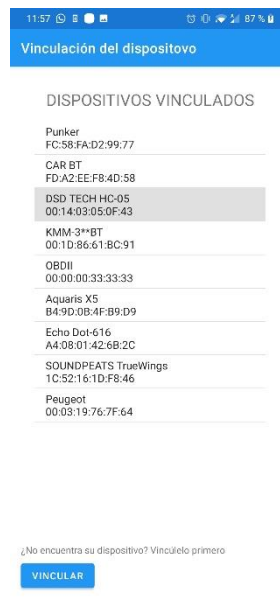


Ilustración 44 Pantalla selección dispositivo Arduino

Recepción de datos

Una vez se ha seleccionado el dispositivo, la siguiente pantalla a mostrar es la de datos, aquí se observa un botón llamado “Recibir datos”, pero antes de pulsarlo es necesario tener todos los sensores ubicados en el dispositivo Arduino

dentro de la sustancia a medir y esperar al menos 5 minutos para que se establezcan correctamente los valores. Tras esta espera y tras pulsar el botón de recibir datos, se mostrará por pantalla un mensaje en la parte inferior informando de que se está estableciendo la conexión y recibiendo los datos (Ilustración 45), si no existe ningún error y no aparece ningún mensaje destinado a tal fin, la información recibida por parte del Arduino se mostrará en pantalla.



Ilustración 45 Pantalla Recepción de datos

Subida de datos

Finalmente, en la misma pantalla anteriormente mostrada, existe un botón llamado “Enviar datos”. Una vez se pulse este botón, la información será enviada a la base de datos (Ilustración 46), quedando finalizado el proceso de la toma de la medición si todo ha ido satisfactoriamente, es por ello que se mostrará un mensaje informando al usuario de esto (Ilustración 47).



Ilustración 46 Pantalla envío de datos en progreso



Ilustración 47 Pantalla envío de datos satisfactorio

Visualización del mapa

Si se pulsa sobre el botón llamado “mapa de mediciones” del menú home (Ilustración 43), el usuario puede visualizar el mapa de mediciones sin necesidad de utilizar un navegador o un ordenador.

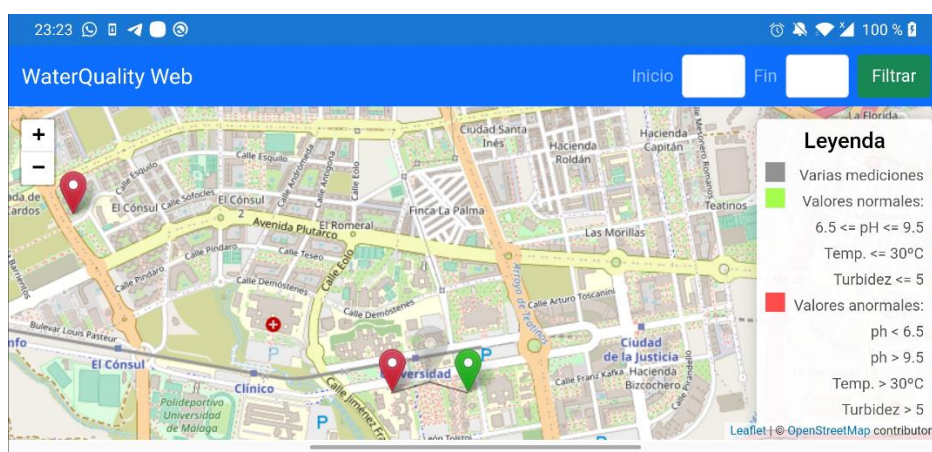


Ilustración 48 Mapa de mediciones. Aplicación móvil

En la parte inferior derecha se muestra una leyenda. Esta leyenda informa al usuario mediante 3 colores el significado de la “chincheta” en el mapa. La “chincheta” gris informa al usuario de que en esta ubicación se localizan varias mediciones. La “chincheta” verde indica que los valores de la medición del agua indican salubridad según lo dispuesto en diferentes leyes y recomendaciones explicadas en la sección del estudio de los parámetros del

agua. La “chincheta” roja indica valores de mediciones fuera del rango de salubridad impuesto.

Manual de la Aplicación web

Visualizar medición

En la aplicación web, se pueden observar todas las mediciones que se han realizado y además se puede ver también todas las mediciones de toda una comunidad. En este caso la interacción es muy sencilla, simplemente aparecerá un mapa con distintas “chinchetas” y simplemente pulsando en una de ellas se podrá ver la información de la medición.

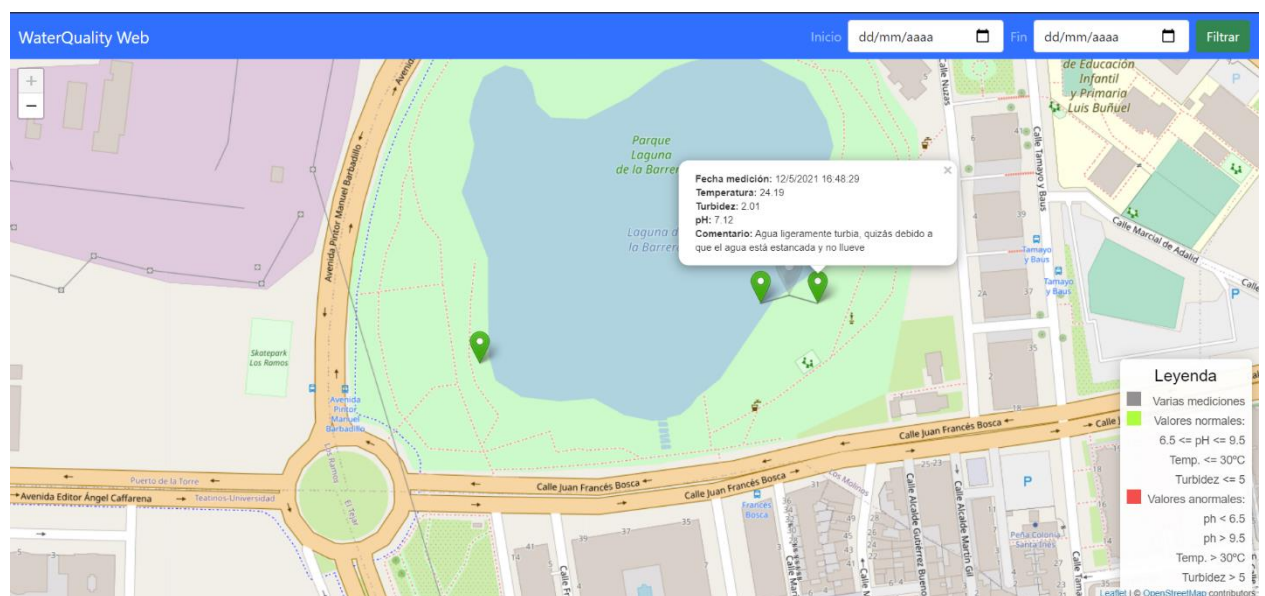


Ilustración 49 Visualizar medición

Filtrar mediciones

La siguiente funcionalidad que tiene la aplicación es la de poder filtrar las mediciones en el mapa en función de la fecha que fue tomada.

Es por ello que en la parte superior se dispone de dos cuadros de texto en los que introducir estas fechas, además si se pulsa el símbolo del calendario, se desplegará un pequeño calendario para elegir la fecha que se desee. Una vez se pulse el botón “filtrar”, aparecerán únicamente las mediciones seleccionadas.

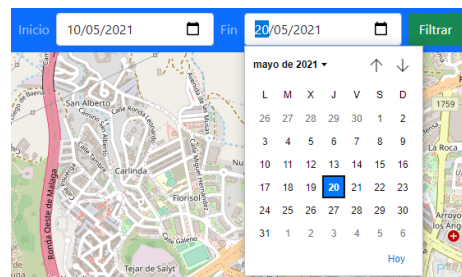


Ilustración 50 Filtrar mediciones

Apéndice C

Coste del prototipo

El coste total de prototipo ha sido de 55,97€. En la tabla 18 se realiza un desglose del precio de cada uno de los componentes.

| Módulo | Precio |
|------------------------------------|---------------|
| Arduino Mega 2560 R3 | 10,96 € |
| Sensor temperatura DFROBOT DS18B20 | 6,72€ |
| Sensor turbidez DFROBOT SEN0189 | 8,32€ |
| Sensor pH DFROBOT SEN0161 | 24,80€ |
| Módulo Bluetooth HC-05 | 1,65€ |
| Protoboard + Conexiones | 3,52€ |
| Total | 55,97€ |

Tabla 18 Coste del prototipo



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga